

ТЕХНИЧЕСКИЕ НАУКИ

КРАТКАЯ ХАРАКТЕРИСТИКА СЕТЕВЫХ КОМАНД ОПЕРАЦИОННОЙ СИСТЕМЫ WINDOWS

Кириченко Александр Аполлонович

net accounts

По этой команде обновляется учётная база пользователей, изменение паролей и параметров подключения для всех пользователей

net computer

По этой команде производится добавление или удаление имени ЭВМ в базе данных домена (вся сеть делится на домены, а домены - на рабочие группы). Команда доступна только для ЭВМ, работающих под управлением Windows 2000/XP Server.

```
C:\>net help computer
Синтаксис данной команды:

NET COMPUTER \имя_компьютера {/ADD | /DEL}

NET COMPUTER - эта команда добавляет или удаляет компьютеры из базы
данных домена, и используется только на серверах Windows NT Server.

\имя_компьютера    Указывает компьютер, который нужно добавить к домену или
                    удалить из домена.
/ADD                Добавляет указанный компьютер к домену.
/DEL                Удаляет указанный компьютер из домена.
```

Справка по команде net computer.

net config

По этой команде выводятся сведения о настраиваемых службах или изменяется настройка служб. Перечень запущенных служб операционной системы можно получить по команде net start. В число служб входят и такие, как «сервер» (в учебных лабораториях обычно защищена и не позволяет получить по ней какую-либо информацию) и «рабочая станция». При отсутствии параметров у этой команды выводятся сведения о настраиваемых службах.

```
C:\>net help config
Синтаксис данной команды:

NET CONFIG [SERVER ; WORKSTATION]

NET CONFIG отображает информацию о настройке служб рабочей станции или
сервера. Когда эта команда используется без указания переключателя SERVER
или WORKSTATION, то выводится список настраиваемых служб. Для того, чтобы
получить справку о том, как выполнить настройку конкретной службы, введите
команду HELP CONFIG имя_службы.

SERVER      Отображает информацию о настройке службы сервера.
WORKSTATION Отображает информацию о настройке службы рабочей станции.

NET HELP имя_команды | MORE   просмотр справки по одному экрану за раз.

C:\>net config
Допускается управление следующими выполняющимися службами:

    Сервер
    Рабочая станция

Команда выполнена успешно.

C:\>net config workstation
Имя компьютера                \\1DAB7876B736490
Полное имя компьютера         1dab7876b736490
Имя пользователя             ААК

Активная рабочая станция на

Версия программы              Windows 2000

Домен рабочей станции        WORKGROUP
DNS-имя домена рабочей станции (null)
Домен входа                   1DAB7876B736490

Интервал ожидания открытия СОМ-порта (с)    0
Отсчет передачи СОМ-порта (байт)            16
Таймаут передачи СОМ-порта (мс)            250
Команда выполнена успешно.
```

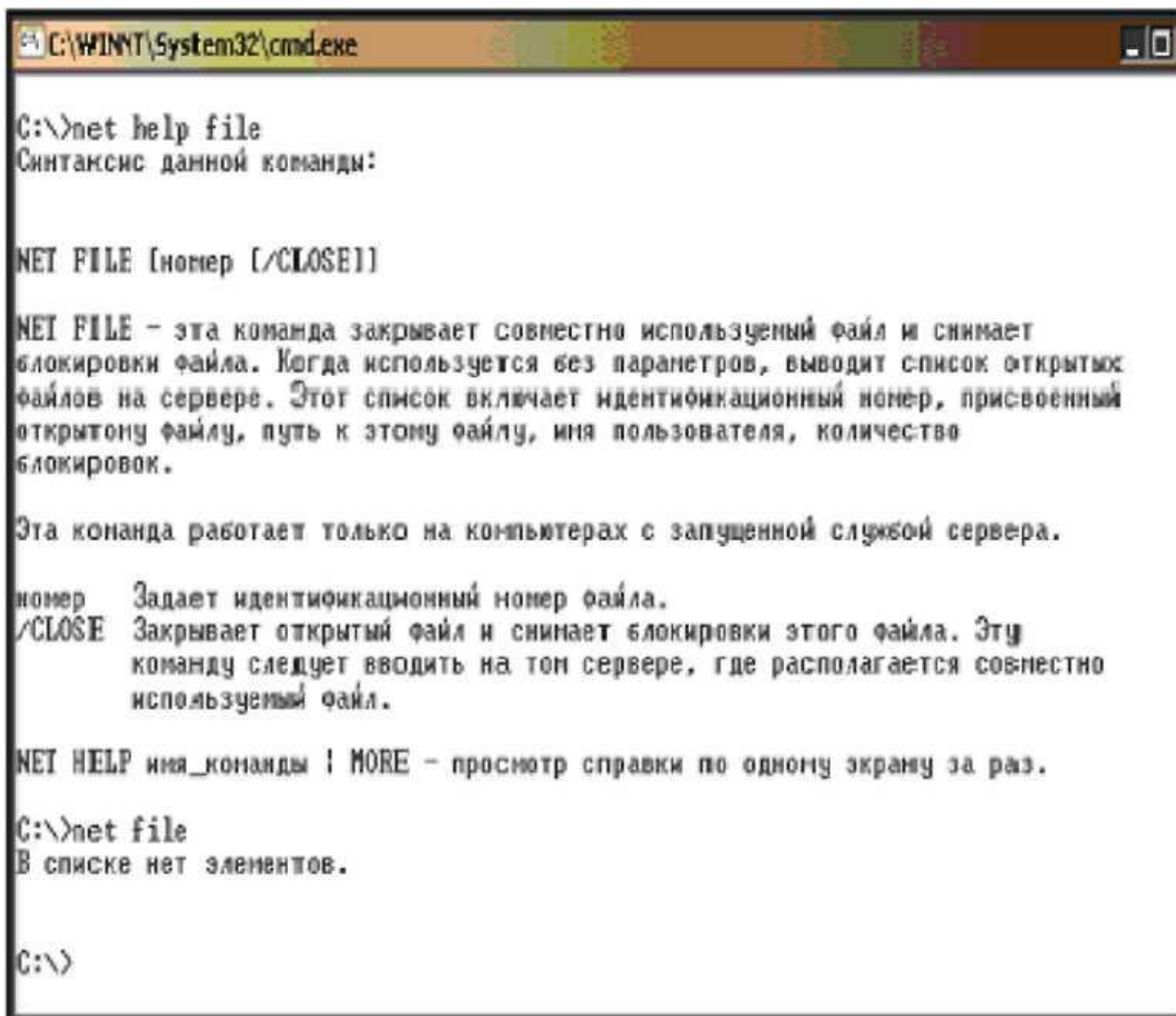
Пример исполнения команд net help config, net config и net config workstation.

net continue

По этой команде производится активация приостановленной службы (имя службы передаётся в качестве параметра).

net file

По этой команде производится вывод имён открытых общих файлов на сервере и количество блокировок для каждого файла, если они установлены



```
C:\WINNT\System32\cmd.exe

C:\>net help file
Синтаксис данной команды:

NET FILE [номер [/CLOSE]]

NET FILE - эта команда закрывает совместно используемый файл и снимает
блокировки файла. Когда используется без параметров, выводит список открытых
файлов на сервере. Этот список включает идентификационный номер, присвоенный
открытому файлу, путь к этому файлу, имя пользователя, количество
блокировок.

Эта команда работает только на компьютерах с запущенной службой сервера.

номер      Задаёт идентификационный номер файла.
/CLOSE     Закрывает открытый файл и снимает блокировки этого файла. Эту
команду следует вводить на том сервере, где располагается совместно
используемый файл.

NET HELP имя_команды ! MORE - просмотр справки по одному экрану за раз.

C:\>net file
В списке нет элементов.

C:\>
```

Справка по команде net help file.

net group

По этой команде производится добавление, вывод и изменение глобальных групп в домене Windows NT/2000/XP Server. Команда доступна только на контроллерах указанного домена. Без параметров команда выводит имя сервера и имена групп на нём.

```
C:\WINNT\System32\cmd.exe

C:\>net help group
Синтаксис данной команды:

NET GROUP [имя_группы [/COMMENT:"текст"]] [/DOMAIN]
        имя_группы {/ADD [/COMMENT:"текст"] | /DELETE} [/DOMAIN]
        имя_группы имя_пользователя [...] {/ADD | /DELETE} [/DOMAIN]

NET GROUP – эта команда добавляет, выводит на экран или изменяет глобальные
группы на сервере. Когда используется без параметров, отображает список
глобальных групп на сервере.

Имя_группы          Задаёт имя группы, которую нужно добавить, расширить или
                    удалить. Для того чтобы получить список пользователей
                    в группе, задайте только имя группы.
/COMMENT:"текст"    Добавляет комментарий для новой или существующей группы.
                    Данная комментарий не должна превышать 48 знаков. Текст
                    комментария должен быть заключен в кавычки.
/DOMAIN             Выполняет операцию на контроллере домена в текущем
                    домене. В противном случае операция выполняется
                    на локальном компьютере.
имя_пользователя [ ... ]
                    Задаёт одно или несколько имен, которые нужно добавить
                    или удалить из группы. Имена пользователей разделяются
                    пробелом.
/ADD                Добавляет группу, или добавляет пользователя в группу.
/DELETE            Удаляет группу, или удаляет пользователя из группы.

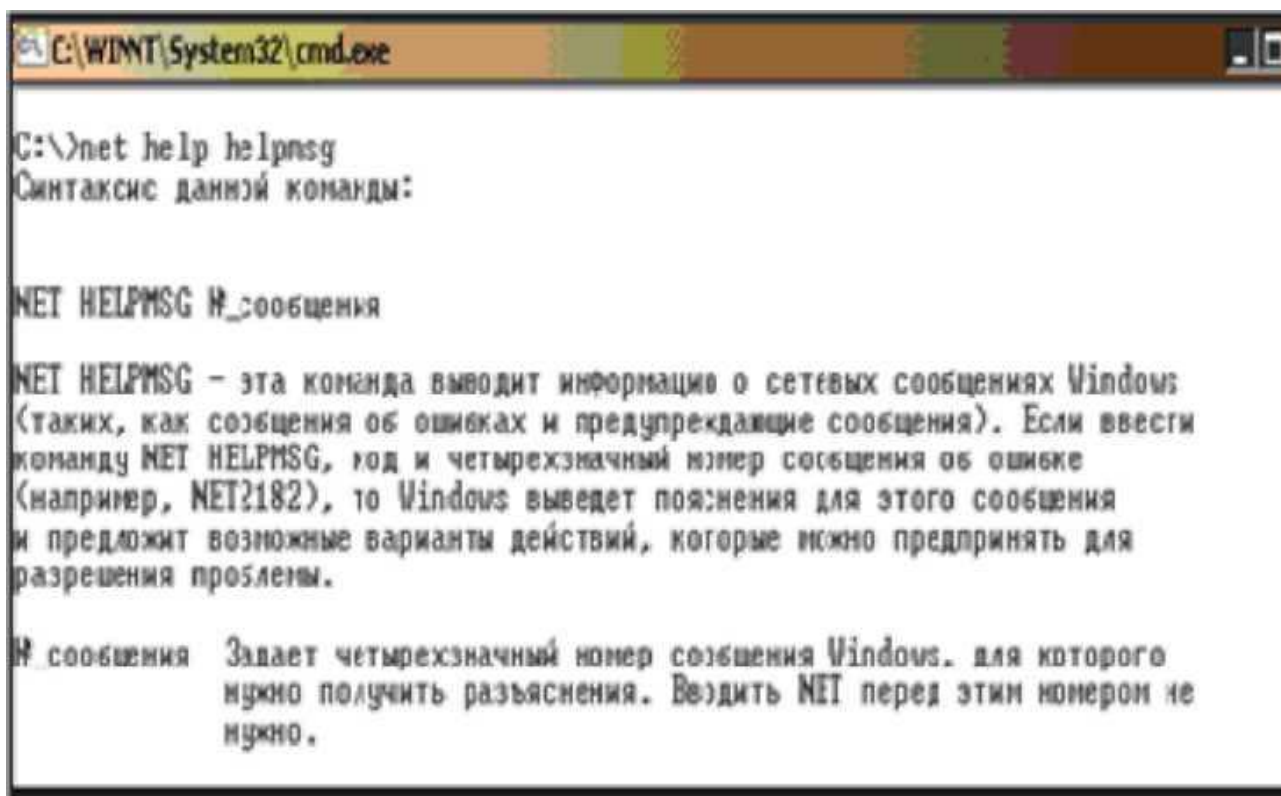
NET HELP имя_команды | MORE – просмотр справки по одному экрану за раз.

C:\>_
```

Справка по команде net group.

net msghelp

По этой команде можно получить справку о сообщениях об ошибках. В качестве параметра обязательно указывается номер сообщения.



```
C:\WINNT\System32\cmd.exe

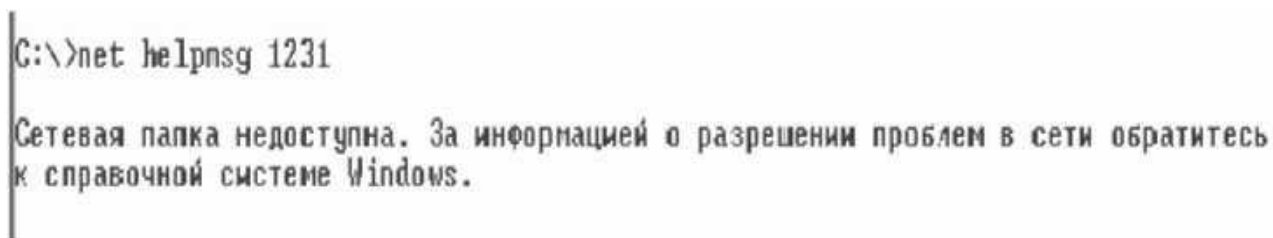
C:\>net help helpmsg
Синтаксис данной команды:

NET HELPMSG №_сообщения

NET HELPMSG – эта команда выводит информацию о сетевых сообщениях Windows
(таких, как сообщения об ошибках и предупреждающие сообщения). Если ввести
команду NET HELPMSG, код и четырехзначный номер сообщения об ошибке
(например, NET?182), то Windows выведет пояснения для этого сообщения
и предложит возможные варианты действий, которые можно предпринять для
разрешения проблемы.

№ сообщения    Задаёт четырехзначный номер сообщения Windows, для которого
                нужно получить разъяснения. Вводить NET перед этим номером не
                нужно.
```

Справка о команде net helpmsg.



```
C:\>net helpmsg 1231

Сетевая папка недоступна. За информацией о разрешении проблем в сети обратитесь
к справочной системе Windows.
```

Справка об ошибке 1231.

net localgroup

По этой команде производится добавление, отображение и изменение локальных групп. Среди параметров могут встречаться имя локальной группы и имя домена. Без параметров команда выводит имя сервера и имена локальных групп на ЭВМ.

```

C:\WINNT\System32\cmd.exe
C:\>net help LOCALGROUP
Синтаксис данной команды:

NET LOCALGROUP [имя_группы [/COMMENT:"текст"] [/DOMAIN]
                имя_группы [/ADD [/COMMENT:"текст"] | /DELETE] [/DOMAIN]
                имя_группы имя [...] [/ADD | /DELETE] [/DOMAIN]

NET LOCALGROUP – эта команда служит для изменения локальных групп на
компьютере. Когда используется без параметров, отображает список локальных
групп на данном компьютере.

имя_группы      Задаст имя локальной группы, которую необходимо добавить,
                расширить или удалить. Если указать только имя группы, то
                будет выведен список пользователей или глобальных групп,
                являющихся членами этой локальной группы.

/COMMENT:"текст" Добавляет комментарий для новой или существующей группы.
                Текст должен быть заключен в кавчки.

/DOMAIN        Выполняет операцию на основном контроллере домена в
                текущем домене. В противном случае операция выполняется
                на локальном компьютере.

имя [...]     Список из одного или нескольких имен пользователей,
                которых необходимо добавить или удалить из локальной группы.
                Имена разделяются пробелом. Эти имена могут быть именами
                пользователей или глобальных групп, но не именами других
                локальных групп. Если пользователь зарегистрирован в другом
                домене, его имени должно предшествовать имя домена
                (например, SALES\RALPHR).

/ADD           Добавляет имя группы или имя пользователя в локальную
                группу. Регистрационная запись для добавляемых пользователей
                или глобальных групп должна быть создана заранее.

/DELETE       Удаляет имя группы или пользователя из локальной группы.

NET HELP имя_команды | MORE - просмотр справки по одному экрану за раз.

C:\>net localgroup

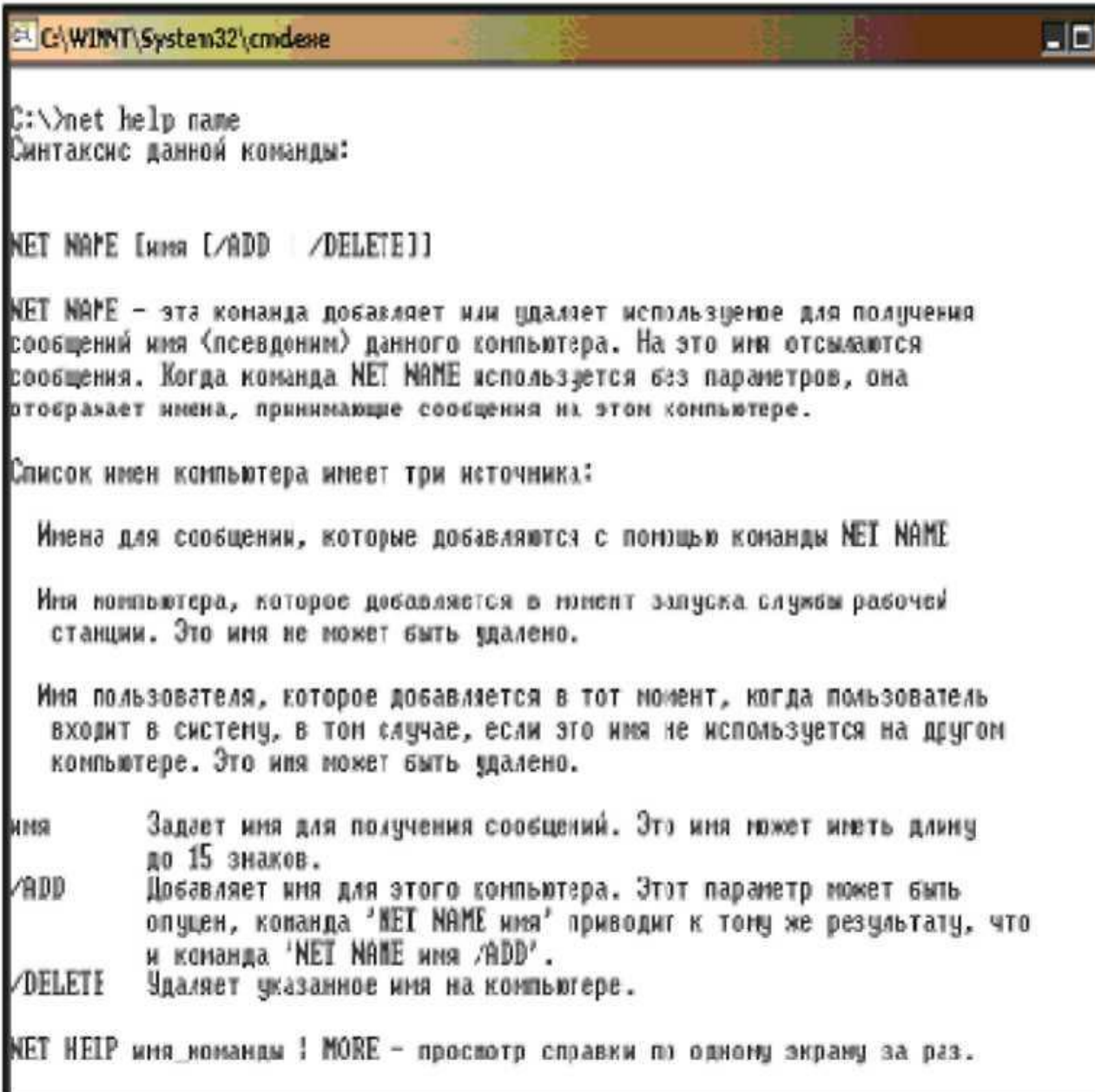
Псевдонимы для \\1DAB7876B731490
-----
*Администраторы          *Гости                   *Операторы архива
*Опытные пользователи   *Пользователи           *Репликатор
Команда выполнена успешно.

C:\>
    
```

Результат выполнения команд net help localgroup и net localgroup.

net name

по этой команде производится добавление или удаление имени, иногда называемого псевдонимом. Это имя используется для вывода сообщений. По этой же команде можно вывести список имён, для которых ЭВМ принимает сообщения. Команда работает только если запущена служба сообщений.



```
C:\WINNT\System32\cmd.exe

C:\>net help name
Синтаксис данной команды:

NET NAME [имя [/ADD | /DELETE]]

NET NAME – эта команда добавляет или удаляет использование для получения
сообщений имя (псевдоним) данного компьютера. На это имя отсылаются
сообщения. Когда команда NET NAME используется без параметров, она
отображает имена, принимающие сообщения на этом компьютере.

Список имен компьютера имеет три источника:

Имена для сообщения, которые добавляются с помощью команды NET NAME

Имя компьютера, которое добавляется в момент запуска службы рабочей
станции. Это имя не может быть удалено.

Имя пользователя, которое добавляется в тот момент, когда пользователь
входит в систему, в том случае, если это имя не используется на другом
компьютере. Это имя может быть удалено.

имя      Задает имя для получения сообщений. Это имя может иметь длину
до 15 знаков.
/ADD     Добавляет имя для этого компьютера. Этот параметр может быть
опущен, команда 'NET NAME имя' приводит к тому же результату, что
и команда 'NET NAME имя /ADD'.
/DELETE  Удаляет указанное имя на компьютере.

NET HELP имя_команды ! MORE – просмотр справки по одному экрану за раз.
```

Справка по команде net name.

net pause

По этой команде приостанавливается работающая служба, имя которой указано в команде в качестве параметра.

```
C:\WINNT\System32\cmd.exe

C:\>net help pause
Синтаксис данной команды:

NET PAUSE имя_службы

NET PAUSE - эта команда приостанавливает службу Windows или ресурс.

имя_службы      Это имя приостанавливаемой службы. Может быть одним
                из следующих имен:

                LPDSVC      <Сервер сетевой печати TCP/IP>
                MACFILE     <Файл-сервер для Macintosh>
                NETDDE      <Служба сетевого DDE>
                NETDDESDM   <Диспетчер сетевого DDE>
                NETLOGON    <Вход в сеть>
                NTLMSSP     <Поддержка безопасности NT LM>
                REMOTEACCS  <RAS-сервер>
                SCHEDULE    <Служба расписания>
                SERVER      <Сервер>
                SIMPTCP     <Службы простого TCP/IP>
                WORKSTATION <Рабочая станция>

NET HELP имя_команды / MORE - просмотр справки по одному экрану за раз.
```

Справка по команде net pause.

net print

По этой команде ведётся управление заданиями и очередями принтеров, и вывод сведений о них.

```

C:\WINNT\System32\cmd.exe
C:\>net help print
Синтаксис данной команды:

NET PRINT \\\имя_компьютера\имя_ресурса
        [\\имя_компьютера] #_задания [/HOLD | /RELEASE | /DELETE]

NET PRINT - эта команда отображает список заданий для печати в совместно
используемых очередях. Для каждой очереди отображается список заданий с
указанием размера и статуса каждого задания, и статус очереди.

\\имя_компьютера      Задает имя компьютера, на котором находятся
совместно используемые очереди заданий на печать.
имя_ресурса          Задает имя совместно используемой очереди принтера.
#_задания             Задает идентифицирующий номер, присвоенный заданию
на печать. Компьютер, на котором находятся одна или
несколько очередей принтеров, присваивает каждому
заданию уникальный номер.
/HOLD                 Задерживает задание в очереди, предотвращая печать.
Задание остается в очереди принтера, другие задания
обходят его, пока оно не будет освобождено.
/RELEASE              Вновь активизирует задержанное ранее задание.
/DELETE              Удаляет задание из очереди.

NET HELP имя_команды | MORE - просмотр справки по одному экрану за раз.

```

Справка по команде net print.

net send

По этой команде производится отправка сообщения другому пользователю, компьютеру или псевдониму в сети. Для получения сообщений на ЭВМ-получателе должна быть запущена служба сообщений.

```

C:\WINNT\System32\cmd.exe
C:\>net help send
Синтаксис данной команды:

NET SEND <имя | * | /DOMAIN[:имя] | /USERS> сообщение

NET SEND - эта команда отправляет сообщения другим пользователям,
компьютерам или иным именам для получения сообщений в сети. Для того, чтобы
получить сообщение, должна быть запущена служба сообщений (MESSENGER).

Отправить сообщение на конкретное имя можно только в том случае, если это
имя активно в сети. Если сообщение отсылается на имя пользователя, то этот
пользователь должен к этому моменту войти в сеть и запустить службу
сообщений для того, чтобы получить это сообщение.

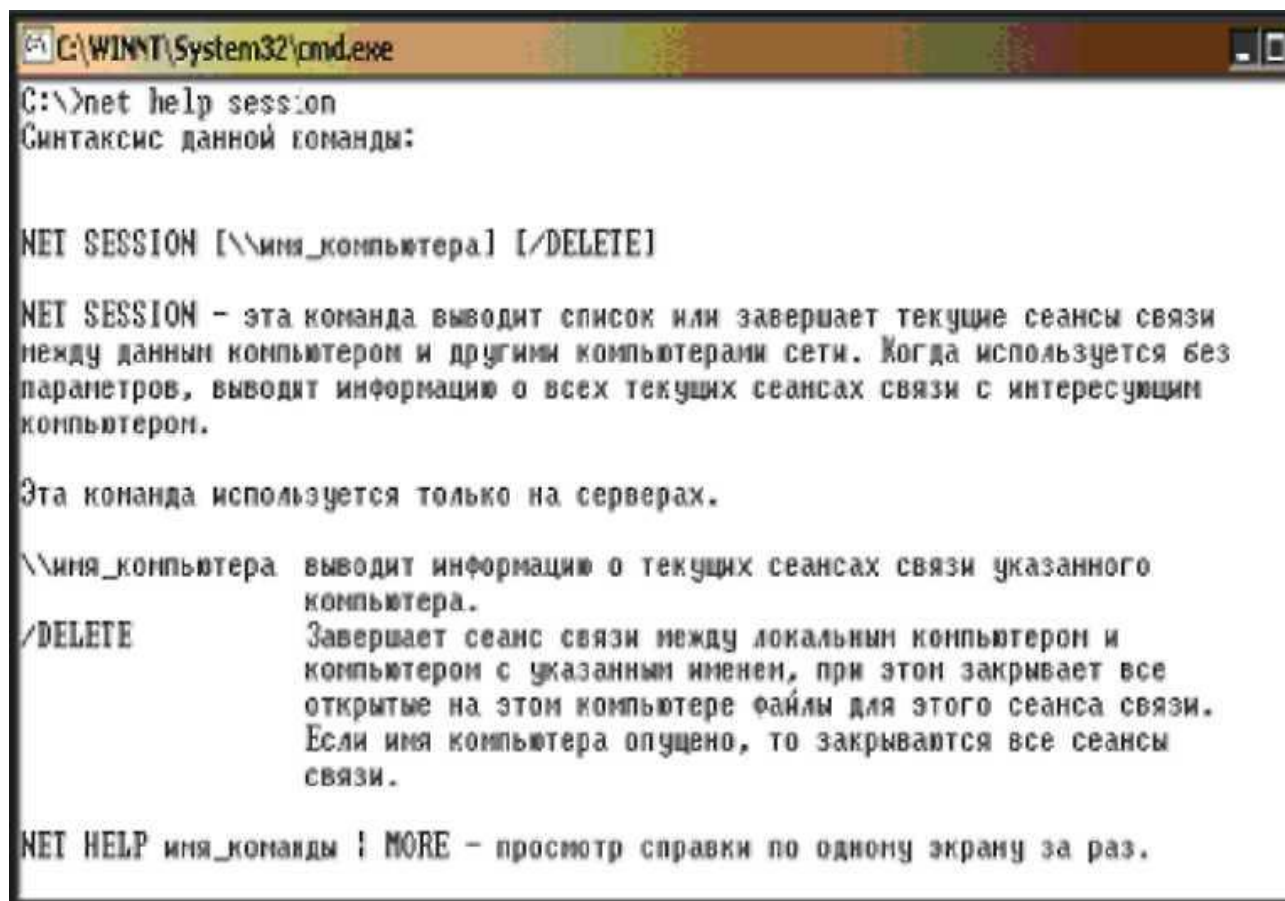
имя                Задаёт имя пользователя, компьютера или имя для
                   получения сообщений, на которое отправляется данное
                   сообщение. Если это имя компьютера, которое содержит
                   пробелы, то оно должно быть заключено в кавычки (" ").
*                  Используется для отправки сообщения по всем именам
                   в текущей группе.
/DOMAIN[:имя]     Направляет сообщение по всем именам домена данной
                   рабочей станции. Если указано имя, то сообщение
                   отправляется по всем именам указанного домена или
                   рабочей группы.
/USERS            Направляет сообщение всем пользователям, подключенным
                   в настоящий момент к серверу.
сообщение         Представляет собой текст отправляемого сообщения.

NET HELP имя_команды | MORE - просмотр справки по одному экрану за раз.
    
```

Справка по команде net send.

net session

По этой команде производится отключение пользователей, подсоединённых к ЭВМ и выводится список подсоединённых пользователей.



```
C:\WINNT\System32\cmd.exe
C:\>net help session
Синтаксис данной команды:

NET SESSION [\\имя_компьютера] [/DELETE]

NET SESSION – эта команда выводит список или завершает текущие сеансы связи
нежду данным компьютером и другими компьютерами сети. Когда используется без
параметров, выводит информацию о всех текущих сеансах связи с интересующим
компьютером.

Эта команда используется только на серверах.

\\имя_компьютера выводит информацию о текущих сеансах связи указанного
компьютера.
/DELETE Завершает сеанс связи между локальным компьютером и
компьютером с указанным именем, при этом закрывает все
открытые на этом компьютере файлы для этого сеанса связи.
Если имя компьютера опущено, то закрываются все сеансы
связи.

NET HELP имя_команды ! MORE – просмотр справки по одному экрану за раз.
```

Справка по команде net session

net share

По этой команде производится создание или удаление совместно используемых ресурсов, выводятся сведения обо всех совместно используемых ресурсах на локальной ЭВМ.

```

C:\WINNT\system32\cmd.exe

C:\>net help share
Синтаксис данной команды:

NET SHARE имя_ресурса
        имя_ресурса=диск:путь [/USERS:число | /UNLIMITED]
                                [/REMARK:"текст"]
                                [/CACHE:Manual | Automatic | No ]
        имя_ресурса [/USERS:число | /UNLIMITED]
                                [/REMARK:"текст"]
                                [/CACHE:Manual | Automatic | No ]
        <имя_ресурса | имя_устройства | диск:путь> /DELETE

NET SHARE – эта команда разрешает использовать ресурсы другим пользователям
в сети. Когда используется без параметров, выводит информация обо всех
ресурсах данного компьютера, которые могут быть совместно использованы.
Для каждого ресурса Windows NT выводит имя устройства или путь и
соответствующий комментарий.

имя_ресурса           Задает сетевое имя данного совместно используемого
ресурса. Если ввести в качестве параметра только имя
ресурса, то выводится информация об этом ресурсе.
диск:путь             Указывает абсолютный путь к совместно используемому
каталогу.
/USERS:число         Устанавливает максимальное число пользователей, которые
могут одновременно получить доступ к совместно
используемому ресурсу.
/UNLIMITED           Определяет, что ограничения на число пользователей,
которые могут получить доступ к совместно используемому
ресурсу, отсутствует.
/REMARK:"текст"     Задает краткое примечание, описывающее ресурс.
Текст должен быть заключен в кавычки.
имя_устройства       Задает один или несколько принтеров <от LPT1: до LPT9:>
совместно используемых под данным именем ресурса.
/DELETE              Прекращает совместное использование данного ресурса.
/CACHE:Automatic     Включает автономное клиентское кэширование с
автоматическим обновлением
/CACHE:Manual        Включает автономное клиентское кэширование с
обновлением, выполняемым вручную
/CACHE:No            Игнорирует клиента, что автономное кэширование
использовать не следует

NET HELP имя_команды ! MORE – просмотр справки по одному экрану за раз.

C:\>net share

Общее имя   Ресурс   Заметки
-----
IPC$        D:\          Удаленный IPC
D$          C:\WINNT    Стандартный общий ресурс
ADMIN$      C:\          Удаленный Admin
C$          C:\          Стандартный общий ресурс
E$          E:\          Стандартный общий ресурс
Когда выполнена успешно.
    
```

Справка по команде net share и результат её использования.

net start

По этой команде производится запуск службы или вывод списка запущенных служб. Имена служб, состоящие из нескольких слов, заключаются в кавычки.

```

L:\WINNT\System32\cmd.exe

C:\>net help start
Синтаксис данной команды:

NET START [служба]

NET START - эта команда выводит список запущенных служб.

Для того, чтобы получить дополнительную справку по поводу конкретной
службы, используйте справочник команд (файл NTCMDS.HLP).

[служба] - является одной из следующих служб:

ALERTER           (Служба оповещения)
BROWSER          (Обозреватель сети)
CLIPSRV          (Сервер буфера обмена)
DHCP             (DHCP-клиент)
EVENTLOG         (Журнал событий)
LMHOSTS          (Поддержка NetBIOS через TCP/IP)
LPSVC            (Сервер сетевой печати TCP/IP)
MESSENGER        (Служба сообщений)
NETDDE           (Служба сетевого DDE)
NETDDEDSM        (Диспетчер сетевого DDE)
NETLOGON         (Вход в сеть)
NMAGENT          (Агент сетевого монитора)
NTLMSSP          (Поддержка безопасности NT LM)
NWCUKSTATION     (Служба клиента для NetWare)
NWSSAPAGENT      (Агент SAP)
RASMAN           (Диспетчер соединений RAS)
REMOTEACCESS     (RAS-сервер)
REPLICATOR       (Репликация каталогов)
RPCLOCATOR       (Локация RPC)
RPCSS            (Служба RPC)
SCHEDULE         (Служба расписания)
SERVER           (Сервер)
SIMPLIP          (Службы простого TCP/IP)
SNMP             (Служба SNMP)
SPOOLER          (Служба буферизации печати)
UPS              (Служба UPS)
WORKSTATION      (Рабочая станция)

Эти службы имеются только для Windows Server:

MAFILE           (Файл-сервер для Macintosh)
NWCUKSTATION     (Служба шлюза для NetWare)
DHCPSEVER        (DHCP-сервер)
MACPRINT         (Сервер печати для Macintosh)
WINS             (Служба имен WINS)

При запуске из командной строки можно использовать русские названия служб,
при этом они должны быть заключены в кавычки и не допускается изменение
прописных букв на строчные и наоборот. Например, команда
NET START "Обозреватель сети"
запускает службу обозревателя сети.

Команда NET START может также использоваться для запуска служб, не входящих
в состав Windows.
    
```

Справка по команде net start.

```
C:\WINNT\System32\cmd.exe
C:\>net start
Запущены следующие службы Windows 2000:

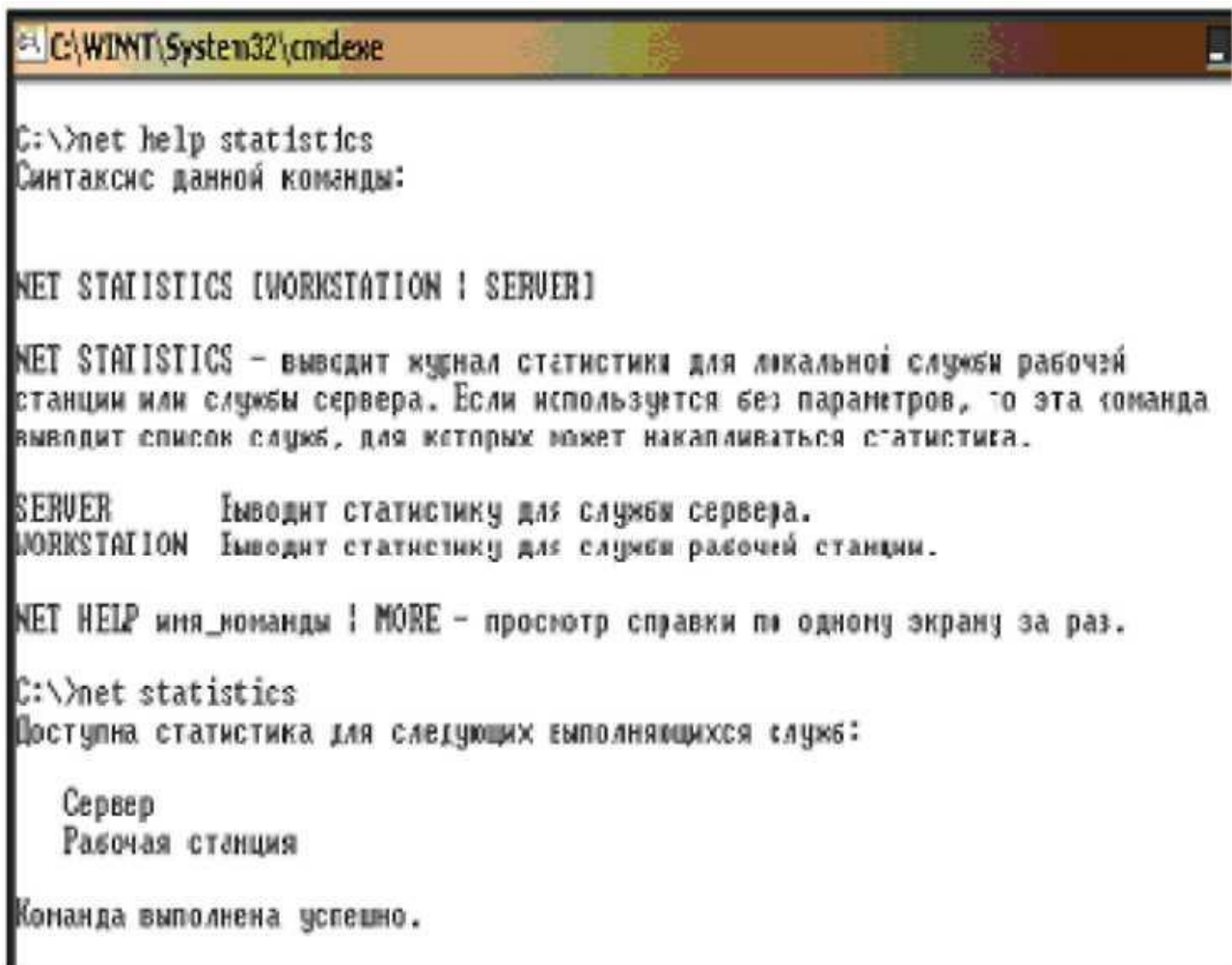
DHCP-клиент
DNS-клиент
Plug and Play
Автоматическое обновление
Агент политики IPSEC
Диспетчер логических дисков
Диспетчер очереди печати
Диспетчер подключений удаленного доступа
Диспетчер учетных записей безопасности
Журнал событий
Защищенное хранилище
Клиент отслеживания изменившихся связей
Обозреватель компьютеров
Планировщик заданий
Рабочая станция
Расширения драйвера оснастки управления Windows
Сервер
Сетевые подключения
Система событий COM+
Служба RunAs
Служба поддержки TCP/IP NetBIOS
Служба сообщений
Служба удаленного управления реестром
Съемные ЗУ
Телефония
Уведомление о системных событиях
Удаленный вызов процедур (RPC)

Команда выполнена успешно.
```

Результат выполнения команды net start без параметров.

net statistics

По этой команде производится вывод журнала статистики для служб локальной рабочей станции или сервера. Команда без параметров позволяет получить список запущенных служб, для которых можно получить статистику. На рисунке приведен пример выполнения команды для рабочей станции.



```
C:\WINNT\System32\cmd.exe

C:\>net help statistics
Синтаксис данной команды:

NET STATISTICS [WORKSTATION | SERVER]

NET STATISTICS - выводит журнал статистики для локальной службы рабочей
станции или службы сервера. Если используется без параметров, то эта команда
выводит список служб, для которых может накапливаться статистика.

SERVER      Выводит статистику для службы сервера.
WORKSTATION Выводит статистику для службы рабочей станции.

NET HELP имя_команды ! MORE - просмотр справки по одному экрану за раз.

C:\>net statistics
Доступна статистика для следующих выполняющихся служб:

    Сервер
    Рабочая станция

Команда выполнена успешно.
```

Справка по команде net statistics и результат её выполнения без параметров.

```

C:\WINNT\System32\cmd.exe
C:\>net statistics workstation
Статистика рабочей станции для \1DAB7876B736490

Статистика после 1/2/2004 10:55 AM

Получено байт                                0
Принятые блоки сообщений сервера SMB        1
Передано байт                                  0
Переданные блоки сообщений сервера SMB      0
Операции чтения                              0
Операции записи                              0
Отказано в чтении                            0
Отказано в записи                            0

Ошибки сети                                  0
Выполненные подключения                     0
Прогнанные подключения                      0
Отключений от сервера                       0

Запущенные сеансы                           0
Зависание сеансов                           0
Сбои в сеансах                              0
Сбои в операциях                            2
Счетчик использования                       0
Счетчик сбоев при использовании            0

Команда выполнена успешно.
    
```

Пример выполнения команды net statistics для рабочей станции.

net stop

По этой команде производится остановка конкретной сетевой службы. В качестве параметра всегда указывается имя останавливаемой службы.

```
C:\WINNT\System32\cmd.exe

C:\>net help stop
Синтаксис данной команды:

NET STOP служба

NET STOP – эта команда останавливает одну из служб Windows.

Остановка одной из служб может привести к отключению сетевых соединений,
используемых этой службой. Кроме того, некоторые службы зависят от других
служб. Остановка одной из служб может привести к остановке других служб.

Вы должны иметь права администратора для того, чтобы остановить службу
сервера. Служба журнала событий (EventLog) не может быть остановлена.

[служба] – является одной из следующих служб:

ALERTER                (Служба оповещения)
BROWSER                (Обозреватель сети)
CLIPSRV                (Сервер буфера обмена)
DHCP                   (DHCP-клиент)
LMHOSTS                (Поддержка NetBIOS через TCP/IP)
LPDSVC                 (Сервер сетевой печати TCP/IP)
MESSENGER              (Служба сообщений)
NETDDE                 (Служба сетевого DDE)
NETDDESDM              (Диспетчер сетевого DDE)
NETLOGON               (Вход в сеть)
NMAGENT                (Агент сетевого монитора)
NTLMSSP                (Поддержка безопасности NT LM)
NWWORKSTATION          (Служба клиента для NetWare)
NWSAPAGENT             (Агент SAP)
RASMAN                 (Диспетчер соединений RAS)
REMOTEACCESS           (RAS-сервер)
REPLICATOR             (Репликация каталогов)
RPCLOCATOR             (Индикатор RPC)
RPCSS                  (Служба RPC)
SCHEDULE               (Служба расписания)
SERVER                 (Сервер)
SIMPTCP                (Службы простого TCP/IP)
SNMP                   (Служба SNMP)
SPOOLER                (Служба буферизации печати)
UPS                    (Служба UPS)
WORKSTATION            (Рабочая станция)

Эти службы имеются только для Windows Server:

MACFILE                (Файл-сервер для Macintosh)
NWWORKSTATION          (Служба шлюза для NetWare)
DHCPSEVER              (DHCP-сервер)
MACPRINT               (Сервер печати для Macintosh)
WINS                   (Служба имен WINS)

Команда NET STOP может также использоваться для остановки служб, не входящих
в состав Windows.

NET HELP имя_команды ; MORE – просмотр справки по одному экрану за раз.
```

Справка по команде net stop.

net time

По этой команде производится синхронизация часов компьютера с часами другого компьютера или домена.

```

C:\WINNT\System32\cmd.exe

C:\>net help time
Синтаксис данной команды:

NET TIME [[\компьютер | /DOMAIN[:домен]] /RTSDOMAIN[:домен]] [/SET]
          [\компьютер] /QUERYSNTP
          [\компьютер] /SEISNTP[:список серверов NTP]

NET TIME синхронизирует показания часов компьютера с другим компьютером
или доменом. Если используется без параметров в домене Windows Server,
выводит текущую дату и время дня, установленные на компьютере, который
назначен сервером времени для данного домена.
Эта команда позволяет задать сервер времени NTP для компьютера.

\компьютер          Задает имя компьютера, который нужно проверить или
                    с которым нужно синхронизировать показания часов.

/DOMAIN[:домен]     Задает домен, с которым нужно синхронизировать
                    показания часов.

/RTSDOMAIN[:домен]  Задает синхронизацию времени с сервером времени
                    (Reliable Time Server) из указанного домена.

/SET                Синхронизирует показания часов компьютера со
                    временем указанного компьютера или домена.

/QUERYSNTP          Отображает назначенный этому компьютеру сервер NTP

/SEISNTP[:ntp server list]
                    Задаст список серверов времени NTP этого компьютера.
                    Это может быть список IP-адресов или DNS-имен,
                    разделенных пробелами. Если задано несколько серверов,
                    список должен быть заключен в кавычки.

NET HELP имя_команды | MORE - просмотр справки по одному экрану за раз.

```

Справка по команде net time.

net use

По этой команде производится подключение к общим сетевым ресурсам или вывод информации о подключённых ЭВМ.

```
NET USE [имя_устройства ; *] [\имя_компьютера\имя_ресурса[\том] [пароль ; *]]  
[/USER: [имя_домена\имя_пользователя]]  
[/USER: [имя_домена_с_точками\имя_пользователя]]  
[/USER: [имя_пользователя@имя_домена_с_точками]]  
[[/DELETE] ; [/PERSISTENT:<YES ; NO>]]
```

```
NET USE <имя_устройства ; *> [пароль ; *] /HOME
```

```
NET USE [/PERSISTENT:<YES ; NO>]
```

NET USE – эта команда подключает компьютер к совместно используемому ресурсу или отключает компьютер от совместно используемого ресурса. Когда используется без параметров, выводит список соединений для данного компьютера.

Синтаксис команды net use.

Параметры команды:

имя_устройства	Назначает имя для подключения к ресурсу или задает устройство, от которого нужно выполнить отключение. Используется два типа имен устройств: дисковые устройства (буквы от D: до Z:) и принтеры (от LPT1: до LPT3:). Если ввести звездочку (*) вместо имени устройства, то назначается следующее незанятое имя.
\\имя_компьютера	Указывает имя компьютера, контролирующего совместно используемый ресурс. Если в имени компьютера используются пробелы, то нужно заключить весь этот параметр в кавычки, вместе с двумя символами обратной косой черты (\\). Длина имени компьютера может быть от 1 до 15 знаков.
имя_ресурса \\volume	Указывает сетевое имя совместно используемого ресурса. Задает тип NetWare на сервере. Для типа, чтобы иметь доступ к серверам NetWare, необходимо установить и запустить службу клиента для NetWare (на Windows Workstation) или службу шлюза для NetWare (на Windows Server).
пароль	Указывает пароль, который нужен для доступа к совместно используемому ресурсу.
*	Вызывает открытие специальной строки ввода пароля. Пароль не выводится на экран во время его ввода в этой строке.
/USER	Указывает другое имя пользователя, с помощью которого устанавливается соединение.
имя_домена	Указывает другой домен. Если указание домена опущено, то подразумевается текущий домен, использовавшийся при входе в сеть.
имя_пользователя	Указывает имя пользователя для входа в сеть.
/HOME	Подключает пользователя к его домашнему каталогу.
/DELETE	Разрывает сетевое соединение и удаляет его из списка постоянных соединений.
/PERSISTENT	Управляет режимом установления постоянных соединений, автоматически подключаемых при входе в систему. По умолчанию используется режим предыдущего соединения.
YES	Запоминает устанавливаемое соединение и обеспечивает его автоматическое подключение при следующем входе в систему.
NO	Не запоминает устанавливаемое соединение или последующие соединения, в результате эти соединения не будут автоматически подключены при следующем входе в систему. Для удаления постоянных соединений используется параметр /DELETE.

Параметры команды net use.

net user

По этой команде производится добавление, редактирование или просмотр учетных сведений пользователя.

```

C:\WINNT\System32\cmd.exe

C:\>net help user
Синтаксис данной команды:

NET USER [имя_пользователя [пароль ; *] [параметры] [/DOMAIN]
        имя_пользователя [пароль ; *] /ADD [параметры] [/DOMAIN]
        имя_пользователя [/DELETE] [/DOMAIN]

NET USER – эта команда создает и изменяет учетные записи пользователей на
компьютере. Когда используется без параметров, выводит список учетных
записей пользователей для данного компьютера. Информация об учетных записях
пользователей хранится в базе данных учетных записей.

Эта команда используется только на серверах.

имя_пользователя    Задает имя пользователя, которое необходимо сбавить,
                    удалить, изменить или вывести на экран. Длина имени
                    пользователя не должна превосходить 20 знаков.
пароль              Назначает или изменяет пароль для учетной записи
                    пользователя. Пароль должен отвечать установленным
                    требованиям на длину – быть не короче, чем значение,
                    установленное параметром /MINPWLEN в команде
                    NET ACCOUNTS, и в то же время не длиннее 14 знаков.
*                  Вызывает открытие специальной строки ввода пароля. Пароль
                    не выводится на экран во время его ввода в этой строке.
/DOMAIN            Выполняет операцию на контроллере домена в
                    текущем домене.
/ADD               Добавляет учетную запись пользователя в базу данных
                    учетных записей.
/DELETE           Удаляет учетную запись пользователя из базы данных
                    учетных записей.
    
```

Синтаксис команды net user.

Параметр	Описание
/ACTIVE:<YES NO>	Активизирует учетную запись или делает ее не активной. Если учетная запись не активна, пользователь не может получить доступ к серверу. По умолчанию используется значение YES (т.е. учетная запись активна).
/COMMENT:"ТЭКСТ"	Добавляет описательный комментарий об учетной записи (длиной не более 48 знаков). Текст должен быть заключен в кавычки.
/COUNTRYCODE:nnn	Использует кодовую страницу нужного языка для вывода справки и сообщений об ошибках. Значение 0 означает выбор кодовой страницы по умолчанию.
/EXPIRES:<дата NEVER>	Устанавливает дату истечения срока действия учетной записи. Если используется значение NEVER, то время действия учетной записи не имеет ограничений срока действия. Дата истечения срока действия задается в формате dd/mm/гг или mm/dd/гг, в зависимости от того, какая кодовая страница используется. Месяц может быть указан цифрами, названием месяца или трехбуквенным его сокращением. В качестве разделителя полей должен использоваться знак косой черты (/).
/FULLNAME:"имя"	Указывает настоящее имя пользователя (а не кодовое имя, заданное параметром имя_пользователя). Настоящее имя следует заключить в кавычки.
/HOMEDIR:путь	Указывает путь к домашнему каталогу пользователя. Этот каталог должен существовать.
/PASSWORDCHG:<YES NO>	Определяет, может ли пользователь изменять свой пароль. По умолчанию используется значение YES (т.е. изменение пароля разрешено).
/PASSWORDREQ:<YES NO>	Определяет, является ли указание пароля обязательным. По умолчанию используется значение YES (т.е. пароль обязателен).
/PROFILEPATH[:путь]	Устанавливает путь к профилю пользователя.
/SCRIPTPATH:путь	Устанавливает расположение пользовательского сценария для входа в систему.

Параметры команды net user.

```

C:\WINNT\System32\cmd.exe
/TYPE:(промежуток | ALL) Устанавливает промежуток времени, во время
которого пользователю разрешен вход в систему.
Этот параметр задается в следующем формате:

    день [-день] [, день [-день]] , время [-время] [, время [-время]]

    Время указывается с точностью до одного часа.
    Дни являются днями недели и могут указываться
    как в полном, так и в сокращенном виде. Время
    можно указывать в 12- и 24-часовом формате.
    Если используется 12-часовой формат, то можно
    использовать am, pm, a.m. или p.m.
    Значение ALL указывает, что пользователь может
    войти в систему в любое время, а пустое
    значение указывает, что пользователь не может
    войти в систему никогда.
    Разделителем полей указания дней недели и
    времени является запятая, разделителем при
    использовании нескольких частей является
    точка с запятой.

/USERCOMMENT:"текст" Позволяет администратору добавлять или изменять
текст комментария к учетной записи.

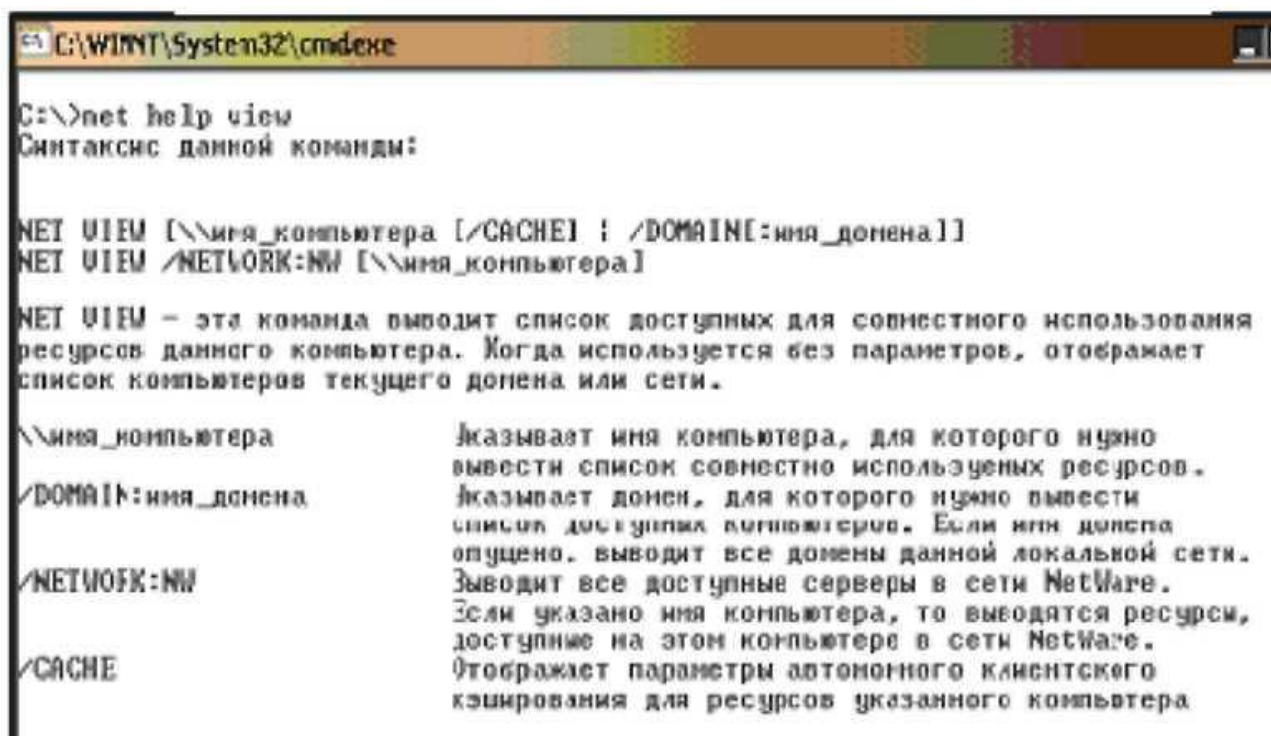
/WORKSTATIONS:(имя_компьютера[,...] | *)
Перечисляет до восьми различных компьютеров,
с которых пользователь может войти в сеть.
Если данный параметр имеет пустой список или
указано значение *, пользователь может войти
в сеть с любого компьютера.

```

Продолжение описания параметров команды net user.

net view

По этой команде производится просмотр списка доменов, ЭВМ или общих ресурсов на указанной ЭВМ.



```
C:\WINNT\System32\cmd.exe

C:\>net help view
Синтаксис данной команды:

NET VIEW [\Имя_компьютера [/CACHE] ; /DOMAIN[:Имя_домена]]
NET VIEW /NETWORK:NW [\Имя_компьютера]

NET VIEW - эта команда выводит список доступных для совместного использования
ресурсов данного компьютера. Когда используется без параметров, отображает
список компьютеров текущего домена или сети.

\Имя_компьютера          указывает имя компьютера, для которого нужно
вывести список совместно используемых ресурсов.
/DOMAIN:Имя_домена       указывает домен, для которого нужно вывести
список доступных компьютеров. Если имя домена
опущено, выводит все домены данной локальной сети.
/NETWORK:NW             выводит все доступные серверы в сети NetWare.
Если указано имя компьютера, то выводятся ресурсы,
доступные на этом компьютере в сети NetWare.
/CACHE                  отображает параметры автономного клиентского
кэширования для ресурсов указанного компьютера
```

Справка по команде net view.

Без параметров команда позволяет вывести список ЭВМ в текущем домене. С помощью этой команды можно вывести список всех имеющихся доменов в сети и общие ресурсы в каждом конкретном домене.

ipconfig

Отображает настройку протокола IP для windows NT/2000/XP.

```

C:\WINNT\System32\cmd.exe
C:\>ipconfig /?

Настройка протокола IP для Windows 2000

ИСПОЛЬЗОВАНИЕ:
ipconfig [/? | /all | /release [адаптер] | /renew [адаптер]
| /flushdns | /registerdns | /showclassid адаптер
| /setclassid адаптер [устанавливаемый_код_класса_dhcp] ]

адаптер    Полное имя или имя, содержащие подстановочные знаки "*" и "?" из
            допустимого множества:
            * - любое количество символов, ? - один любой символ.

ключи:
/?          Отобразить это справочное сообщение.
/all        Отобразить полную информацию о настройке параметров.
/release    Освободить IP-адрес для указанного адаптера.
/renew      Обновить IP-адрес для указанного адаптера.
/flushdns   Очистить кэш разрешений DNS.
/registerdns Обновить все DHCP-аренды и перерегистрировать DNS-имена.
/displaydns Отобразить содержимое кэша разрешений DNS.
/showclassid Отобразить все допустимые для этого адаптера коды (ID)
            классов DHCP.
/setclassid Изменить код класса DHCP (ID).

По умолчанию отображается только IP-адрес, маска подсети и стандартный шлюз
для каждого подключенного адаптера, для которого выполнена привязка с
TCP/IP.

Для ключей /Release и /Renew, если не указано имя адаптера, то будет
освобожден или обновлен IP-адрес, выданный для всех адаптеров,
для которых существуют привязки с TCP/IP.

Для ключа SetClassID, если не указан код класса (ID),
то существующий код класса будет удален.

Примеры:
> ipconfig                - Отображает краткую информацию.
> ipconfig /all           - Отображает полную информацию.
> ipconfig /renew         - Обновляет сведения для всех адаптеров.
> ipconfig /renew EL*     - Обновляет сведения для адаптеров,
                            начинающихся с EL....
> ipconfig /release *ELINK?21* - Освобождает IP-адреса для всех адаптеров,
                            удовлетворяющих запросу, например, ELINK-21, myELELINKi21adapter.

```

Справка по команде ipconfig

СОСТАВ КОМАНД WINDOWS

Кириченко Александр Аполлонович

Команды справочной системы

HELP	Выводит справочную информацию о системе команд Windows2000/XP.
HELP имя_команды	Выводит справочную информацию для набранной команды
Имя_команды /?	Выводит справочную информацию для набранной команды

Команды файловой системы

ATTRIB	Отображение и изменение атрибутов файлов.
CD	Смена текущей папки.
CHDIR	Вывод имени либо смена текущей папки.
CHKDSK	Проверка диска и вывод статистики.
CLS	Очистка экрана.
COMP	Сравнение содержимого двух файлов или двух наборов файлов.
COPY	Копирование одного или нескольких файлов в другое место.
DEL	Удаление одного или нескольких файлов.
DIR	Вывод списка файлов и подпапок из указанной папки.
DISKCOMP	Сравнение содержимого двух гибких дисков
DISKCOPY	Копирование содержимого одного гибкого диска на другой.
ERASE	Удаление одного или нескольких файлов.
FC	Сравнение двух файлов или двух наборов файлов и вывод различий между ними.
FIND	Поиск текстовой строки в одном или нескольких файлах.
FINDSTR	Поиск строк в файлах.
FORMAT	Форматирование диска для работы с Windows2000/XP.
LABEL	Создание, изменение и удаление меток тома для дисков.
MD	Создание папки.
MKDIR	Создание папки.
MOVE	Перемещение одного или нескольких файлов из одной папки в другую.
PUSHD	Сохранение значения текущей активной папки и переход к другой папке.
POPD	Восстановление предыдущего значения текущей активной папки, сохраненного с помощью команды PUSHD.
PRINT	Вывод на печать содержимого текстовых файлов.
RD	Удаление папки.
REN	Переименование файлов и папок.
RENAME	Переименование файлов и папок.
REPLACE	Замещение файлов.
RMDIR	Удаление папки.
SORT	Сортировка ввода.
TREE	Графическое отображение структуры папок заданного диска или заданной папки.

TYPE	Вывод на экран содержимого текстовых файлов.
VERIFY	Установка режима проверки правильности записи файлов на диск
VOL	Вывод метки и серийного номера тома для диска.
XCOPY	Копирование файлов и дерева папок.

Команды управления работой ОС

ASSOC	Вывод либо изменение сопоставлений по расширениям имен файлов.
AT	Выполнение команд и запуск программ по расписанию.
BREAK	Включение/выключение режима обработки комбинации клавиш CTRL+C.
CACLS	Отображение/редактирование списков управления доступом (ACL) к файлам.
CHCP	Вывод либо установка активной кодовой страницы.
CHKNTFS	Отображение или изменение выполнения проверки диска во время загрузки.
CMD	Запуск еще одного интерпретатора командных строк Windows2000/XP.
COLOR	Установка цвета текста и фона, используемых по умолчанию.
COMPACT	Отображение/изменение сжатия файлов в разделах NTFS.
CONVERT	Преобразование дисковых томов FAT в NTFS.
DATE	Вывод либо установка текущей даты.
DOSKEY	Редактирование и повторный вызов командных строк; создание макросов.
FTYPE	Вывод либо изменение типов файлов, используемых при сопоставлении по расширениям имен файлов.
GRAFTABL	Позволяет Windows2000/XP отображать расширенный набор символов в графическом режиме.
MODE	Конфигурирование системных устройств.
MORE	Последовательный вывод данных по частям размером в один экран.
PATH	Вывод либо установка пути поиска исполняемых файлов.
PROMPT	Изменение приглашения в командной строке Windows2000/XP.
RECOVER	Восстановление читаемой информации с плохого или поврежденного диска.
SET	Вывод, установка и удаление переменных среды Windows2000/XP.
START	Запуск программы или команды в отдельном окне.
SUBST	Сопоставляет заданному пути имя диска.
VER	Вывод сведений о версии Windows2000/XP.

Команды пакетных файлов

CALL	Вызов одного пакетного файла из другого.
ECHO	Вывод сообщений и переключение режима отображения команд на экране.
ENDLOCAL	Конец локальных изменений среды для пакетного файла.
EXIT	Завершение работы программы CMD.EXE (интерпретатора командных строк).
FOR	Запуск указанной команды для каждого из файлов в наборе.
GOTO	Передача управления в отмеченную строку пакетного файла.
IF	Оператор условного выполнения команд в пакетном файле.
PAUSE	Приостановка выполнения пакетного файла и вывод сообщения.
REM	Помещение комментариев в пакетные файлы и файл CONFIG.SYS.
SETLOCAL	Начало локальных изменений среды для пакетного файла.

SHIFT	Изменение содержимого (сдвиг) подставляемых параметров для пакетного файла.
-------	---

Сетевые команды Windows 2000/XP

Операционная система WindowsXP позволяет работать с сетями двух типов: с одноранговыми (т.е. распределёнными) сетями, и с сетями типа «клиент - сервер». При работе с обоими типами сетей используется стандартная система команд операционной системы. Но есть и группа специальных, сетевых команд. Большинство из них начинается со слова «net».

Список всех доступных команд группы net можно получить, набрав в консоли команд:

```
net/?
```

На рис. 1 приведена справка по структуре команды net.

```
C:\>net /?
Синтаксис данной команды:

NET [ ACCOUNTS | COMPUTER | CONFIG | CONTINUE | FILE | GROUP | HELP |
  HELPMMSG | LOCALGROUP | NAME | PAUSE | PRINT | SEND | SESSION |
  SHARE | START | STATISTICS | STOP | TIME | USE | USER | VIEW ]

C:\>_
```

Рис. 1. Синтаксис команды net.

Справку по конкретной команде можно получить, набрав после названия группы команд имя конкретной команды (рис. 2).

```
C:\>net accounts /?
Синтаксис данной команды:

NET ACCOUNTS [/FORCELOGOFF:<минуты | NO>] [/MINPWLEN:длина]
  [/MAXPWAGE:<дни | UNLIMITED>] [/MINPWAGE:дни]
  [/UNIQUERPW:число] [/DOMAIN]
```

Рис. 2. Справка по команде netaccounts.

Более подробную справку по конкретной команде можно получить, набрав:

```
net help accounts (рис. 3).
```



```

C:\WINNT\System32\cmd.exe
C:\>NET HELP accounts /?
Синтаксис данной команды:

NET ACCOUNTS [/FORCELOGOFF:(минуты | NO)] [/MINPWLEN:длина]
              [/MAXPWAGE:(дни | UNLIMITED)] [/MINPWAGE:дни]
              [/UNIQUERPW:число] [/DOMAIN]

NET ACCOUNTS - эта команда используется для обновления базы данных
регистрационных записей и изменения параметров входа в сеть (LOGON)
и требований к паролям для всех регистрационных записей.
При использовании этой команды без указания параметров NET ACCOUNTS
выводит текущие значения параметров, определяющих требования к паролям и
входу в сеть, а также информацию о домене.

Должны быть выполнены два условия для того, чтобы изменения параметров
с помощью команды NET ACCOUNTS вступили в силу:

Требования к паролям и параметрам входа в сеть можно применять только
в том случае, если были определены регистрационные записи пользователей
(с помощью Диспетчера пользователей или команды NET USER).

На всех серверах домена, проверяющих полномочия при входе в сеть,
должна быть запущена служба входа в сеть. Эта служба запускается
автоматически при запуске Windows.

/FORCELOGOFF:(минуты | NO)   Устанавливает время в минутах, через которое
пользователь будет принудительно отключен по
истечении срока действия его регистрационной
записи или разрешенного интервала времени.
По умолчанию используется значение NO, т.е.
принудительное отключение не используется.

/MINPWLEN:длина              Устанавливает минимальное количество знаков,
которое должен иметь пароль. Допустимый
диапазон значений: 0-14 знаков, по умолчанию
используется значение 6.

/MAXPWAGE:(дни | UNLIMITED) Устанавливает максимальный срок жизни пароля
(в днях). Для указания бессрочного действия
пароля используется значение UNLIMITED.
Значение параметра /MAXPWAGE не может быть
меньше /MINPWAGE.
Допустимый диапазон значений: 1-999 дней;
по умолчанию используется 90 дней.

/MINPWAGE:дни                Устанавливает минимальный срок жизни пароля
(в днях), по истечении которого пользователь
может изменить пароль. Значение 0 позволяет
менять пароль как угодно часто.
Допустимый диапазон значений: 1-999 дней;
по умолчанию используется 0 дней.
Значение параметра /MINPWAGE не может быть
больше /MAXPWAGE.

/UNIQUERPW:число            Устанавливает требование, чтобы определяемый
пользователем новый пароль не повторял ни
одного из последних использовавшихся ранее
паролей. Максимальное значение - 24.

/DOMAIN                      Выполняет данную операцию на контроллере
домена текущего активного домена.

```

Рис. 3. Справка по команде nethelpaccounts(параметр "| more" позволяет выводить длинную справку по частям, размером в один экран).

Сетевые команды в командной строке

Сетевые возможности Windows не ограничиваются одними сетевыми командами, работающими только в консоли команд. Сетевые команды, исполняемые из командной строки, имеют другую организацию (синтаксис). Например, чтобы установить связь с ресурсом на другом компьютере сети, необходимо в командной строке набрать:

```
\\имя_компьютера\ имя_папки_(или другого ресурса)
```

Имя компьютера можно найти в папке "Сетевое окружение" графического интерфейса Windows-там отображаются все ЭВМ, подключенные к сети в данный момент.

Чтобы послать сообщение на другой компьютер, в командной строке нужно набрать:

```
\\имя_машины net send сообщение
```

РОЛЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И РЕСУРСОВ В МАРКЕТИНГЕ

Чефранова Ольга Игоревна

Ужесточение конкуренции заставляет предприятия активнее внедрять информационные технологии в бизнес-процессы. При этом существенно изменяется круг задач, решаемых менеджерами.

Под термином «информационные технологии» понимается совокупность программно-технических средств вычислительной техники, а также приемов, способов и методов их применения для выполнения функций сбора, хранения, обработки, передачи и использования информации в конкретных предметных областях.

Понятие информационных технологий включает большое количество составляющих: аппаратные платформы, операционные системы, языки программирования и средства разработки приложений, сетевые технологии, базы данных и многие другие.

В методическом смысле можно выделить два основных направления применения информационных технологий и систем в бизнесе (рисунок1). Первое связано с применением информационных технологий и инструментария Интернет для поддержки существующего бизнеса и повышения эффективности управления.

На крупных предприятиях для поддержки существующего бизнеса часто разрабатываются и применяются информационные технологии и системы (ИТС) подготовки и принятия управленческих решений. Корпоративные информационные системы (КИС) объединяют программные и аппаратные средства, информационные технологии и ресурсы для эффективного управления предприятием.

Постоянно разрабатываются новые подходы к управлению бизнесом, которые реализуются в информационных системах.



Рисунок 1. Направления применения ИТС для управления и развития бизнеса

ERP- системы (Enterprise Resource Planning - планирование ресурсов предприятия), позволяют объединить основные элементы производственной и коммерческой деятельности (производство, планирование, финансы и бухгалтерию, материально-техническое снабжение, сбыт, управление запасами, ведение заказов на изготовление и поставку продукции, оказание услуг). Для малых и средних предприятий эффективным вложением средств становится внедрение CRM (Customer Relationships Management - управление взаимоотношениями с клиентами) - систем, нацеленных на сохранение заданного уровня продаж.

Наряду с первым направлением (см. рис. 1), информационные технологии позволяют развивать и второе направление, связанное с созданием новых моделей бизнеса. Речь идет, например, об Интернет-магазинах, электронных торговых площадках, взаимодействии с виртуальными информационными агентствами и т.д.

ИТС электронного взаимодействия, ИТС электронной коммерции и все второе направление тесно связаны с применением Интернет-технологий в менеджменте.

Расширяющееся применение информационных технологий и Интернета в менеджменте объясняется, прежде всего, стремлением предприятий увеличить свою прибыль за счет сокращения соответствующих издержек. В организации взаимодействия с клиентами, принятии важнейших решений по формированию ценовой, сбытовой, коммуникативной политики ведущая роль отводится сотрудникам, способным оперативно собирать, систематизировать и обрабатывать информацию об изменениях во внешней и внутренней среде фирмы. Как правило, им поручается разработка и сопровождение содержательной части корпоративного Web-сайта, презентаций. Они участвуют в организации работы на электронных торговых площадках, проводят маркетинговые исследования в сети Интернет, готовят предложения по развитию новых направлений бизнеса.

СЕТЕВЫЕ УТИЛИТЫ ОС WINDOWS

Кириченко Александр Аполлонович

Утилита ping операционной системы Windows NT/2000/XP служит для проверки работоспособности сети. Синтаксис:

ping хост [таймаут]

ping-s[-drvRlfnq] [-i время_ожидания] [-p шаблон] хост [размер_данных [кол_пакетов]]

Описание:

Команда ping использует датаграмму ECHO_REQUEST протокола ICMP, чтобы вызвать ответ ICMP_ECHO_RESPONSE указанного хоста или сетевого шлюза. Если хост отвечает, ping выдает сообщение, что хост жив (хост is alive) в стандартный выходной поток, и завершает работу. В противном случае, после таймаут секунд она выдает сообщение, что от хоста ответа нет (no answer from хост). Стандартное значение таймаута - 20 секунд.

Опции

Команда ping воспринимает следующие опции:

-d	Режим отладки. Поставщику передается опция SODEBUG.
-f	Лавинный ping. Выдает пакеты сразу после возвращения или сто раз в секунду, в зависимости от того, что быстрее. Для каждого посланного ECHOREQUEST печатается точка а для каждого полученного ECHOREPLY печатается забой (backspace). Это позволяет быстро оценить, сколько пакетов потеряно. Только привилегированный пользователь может использовать эту опцию. Эта команда может существенно увеличить загрузку сети и ее надо использовать осторожно.
-i время ожидания	Ожидать время ожидания секунд между посылками пакетов. По умолчанию интервал между посылками пакетов - одна секунда. Эта опция несовместима с опцией -f.
-l	Исключить маршрут к источнику. Использует эту опцию в заголовке IP для посылки пакета указанному хосту и обратно. Обычно указывается с опцией -R. Опция -l допустима только когда в качестве хоста указан localhost или uname-n\
-n	Только числовая выдача. Не предпринимать попытки искать символьные имена для адресов хостов.
-p шаблон	Указанный шаблон используется для заполнения посылаемых пакетов. Шаблон задается, как шестнадцатеричная строка байтов и может иметь длину до 16 байтов. Шаблон повторяется для заполнения раздела данных пакета. Например, -rfff вызывает заполнение пакетов единицами. Эта опция полезна при поиске проблем сети, связанных с передаваемыми данными.
	Сокращенный вывод. Не выдается ничего, кроме суммарных строк при запуске и завершении работы.
-r	Не использовать обычные таблицы маршрутизации и посылать напрямую указанному хосту в подключенной сети. Если хост не находится в непосредственно подключенной сети, возвращается ошибка. Эту опцию можно использовать для обращения к локальному хосту через интерфейс, удаленный демоном маршрутизации (см. routed(1M)).

-R	Записать маршрут. Устанавливает опцию записи маршрута IP, в результате чего маршрут пакета будет записан в заголовке IP. Содержимое записи маршрута будет выдано, только если указана опция -vi будет устанавливаться для возвращаемых пакетов, только если целевой хост сохраняет запись маршрута между выдачами или если задана опция -1.
-s	Посылать датаграмму каждую секунду и печатать строку вывода для каждого полученного ответа ECHORESPONSE(если ответа нет, ничего не выдается).
-v	Детальный вывод. Выдает все полученные пакеты ICMP, кроме ECHORESPONSE.

Использование утилиты: Когда указан флаг -s, pingпосылает датаграмму каждую секунду и печатает одну строку вывода для каждого полученного ответа ECHO_RESPONSE. В этом случае pingвычисляет времена обхода (roundtriptimes) и статистику потери пакетов; после завершения или по истечении таймаута команда печатает соответствующую итоговую информацию. Если указано необязательное количество пакетов (кол_пакетов), pingпосылает только соответствующее количество запросов. Если количество пакетов не указано, команда будет выполняться бесконечно. Для прекращения продолжающегося вывода, используется клавиша прерывания (Del).

Стандартный размер пакета датаграммы равен 64 байтам, но можно задать и другой размер с помощью аргумента командной строки размер_данных. Поскольку pingавтоматически добавляет 8- байтовый заголовок к каждой посылаемой датаграмме, размер пакета, показываемый при использовании опции -sc аргументом размер_данных, всегда будет на 8 байтов больше, чем указанное вами значение.

При использовании pingдля поиска сбоев в сети, необходимо сначала выполнить pingна локальный хост, чтобы убедиться, что работает локальный сетевой интерфейс.

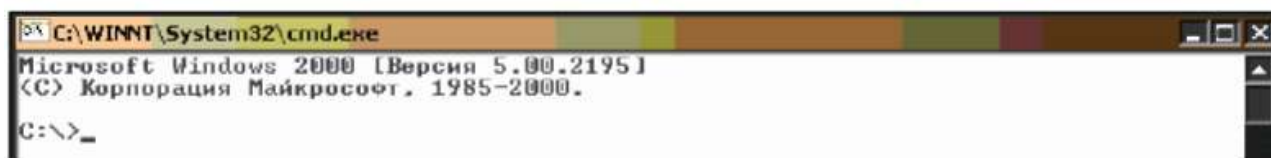
Tracert - Трассировка пути до сервера

Сервис Tracert (от англ. trace route) помогает определить путь прохождения запроса до интересующего Вас сервера и узнать, сколько и какие промежуточные интернет-серверы, маршрутизаторы и другие устройства участвуют в пересылке данных на сервер и обратно. Вы также сможете определить время отклика каждого узла на пути до сервера, что позволит оценить, на каком участке возникают наибольшие задержки, связанные с маршрутизацией и передачей данных. Для уменьшения влияния случайных факторов сервис генерирует несколько запросов, в результате чего выдается минимальное, максимальное и среднее время отклика каждого узла пути.

Данный сервис по сути является техническим инструментом для определения тех узлов сети, которые вносят наибольшие задержки при маршрутизации запросов и может быть использован для определения места расположения искомого узла и взаимосвязи глобальных вычислительных сетей.

Для обращения к данному сервису необходимо:

1. Нажать Пуск - Выполнить - cmd. Появится окно консоли команд:



2. В появившемся окне набрать команду `tracert` и указать в ней адрес сайта, до которого необходимо проследить трассу

```
C:\>tracert -w 5000 www.mit.edu

Трассировка маршрута к DANDELION-PATCH.mit.edu [18.181.0.31]
с максимальным числом прыжков 30:

  1    140 ms    190 ms    120 ms    ts23.Moscow.net.ro1.ru [194.67.8.62]
  2    130 ms    140 ms    140 ms    isl-hub-20.Moscow.gldn.net [194.67.8.9]
  3    170 ms    130 ms    141 ms    cisco02.Moscow.gldn.net [194.186.0.197]
  4    130 ms    140 ms    141 ms    cisco02.Moscow.gldn.net [194.67.17.238]
  5    160 ms    161 ms    150 ms    POS6-2.GW9.STK2.ALTER.NET [146.188.48.27]
  6    161 ms    150 ms    160 ms    so-4-0-0.XR1.STK3.ALTER.NET [146.188.7.77]
  7    171 ms    190 ms    180 ms    ge-0-0-0.IR2.STK2.ALTER.NET [146.188.11.238]
  8    270 ms    271 ms    270 ms    so-2-0-0.IR2.DCA4.ALTER.NET [146.188.11.222]
  9    261 ms    280 ms    271 ms    so-0-0-0.IL2.DCA6.ALTER.NET [146.188.13.41]
 10    271 ms    270 ms    270 ms    0.so-1-0-0.IL2.DCA6.ALTER.NET [152.63.9.222]
 11    280 ms    271 ms    270 ms    0.so-7-0-0.XL2.DCA5.ALTER.NET [152.63.42.189]
 12    270 ms    271 ms    270 ms    0.so-7-0-0.BR1.DCA5.ALTER.NET [152.63.43.177]
 13    270 ms    270 ms    271 ms    so-0-0-0.edge1.Washington1.Level3.net [209.244.2
19.157]
 14    280 ms    280 ms    281 ms    so-2-1-0.bhr2.Washington1.Level3.net [209.244.11
.13]
 15    271 ms    280 ms    280 ms    so-0-1-0.npi.Boston1.Level3.net [209.247.9.125]

 16   1071 ms    281 ms    280 ms    ge-10-1.hs1.Boston1.Level3.net [64.159.4.154]
 17    270 ms    261 ms    270 ms    p0-0.mit3.bbplanet.net [4.24.88.50]
 18    270 ms    270 ms    271 ms    NW12-XIR-2-BACKBONE.MIT.EDU [18.168.0.21]
 19    271 ms    280 ms    281 ms    DANDELION-PATCH.MIT.EDU [18.181.0.31]

Трассировка завершена.
```

3. Если программа не может получить трассу из-за большого времени задержки, увеличить допустимое время с помощью параметра `-w` (например, до 10000).

```
Использование: tracert [-d] [-h максЧисло] [-j списокУзлов] [-w интервал] имя

Параметры:
-d          Без разрешения в имена узлов.
-h максЧисло Максимальное число прыжков при поиске узла.
-j списокУзлов Свободный вывод маршрута по списку узлов.
-w интервал  Интервал ожидания каждого ответа в миллисекундах.
```

Если программа `tracert` недоступна в Вашей сети, можно воспользоваться доступным средством, расположенным в Интернет, например, по адресу <http://www.whois-service.ru/traceroute/>. На этом сервере доступны и другие программы для исследования сайтов:

- *DNS WATCH* - просмотр записей о домене на DNS-серверах, а также информации о серверах, поддерживающих работу зоны DNS и почты для домена
- *WHATS* - Что запущено на сервере? Сервис Whats (от англ. What is running?) позволяет определить, под управлением какого web-сервера работает сайт, а также какая операционная система запущена на данном сервере. Это позволяет ориентироваться на то, какие программные продукты и операционные системы используют ведущие в своих областях проекты фирмы, и воспользоваться опытом существующих сайтов.
- *IP LOOKUP* - информация о том, какой организации принадлежит ip-адрес или подсеть класса А, В либо С, на какой хостинг-площадке размещается сайт

Использование пакета Google для администрирования Web-

САЙТОВ

Используя поисковые машины, такие как Google, можно решить многие задачи администрирования. Нестандартные возможности поисковой машины Google подробно рассмотрены в «*Google Hacks, 2nd Edition, By Tara Calishain, Rael Dornfest, Publisher: O'Reilly, 2004, ISBN: 0-596-00857-0, Pages: 479*».

Частичный перевод книги можно найти по адресу: <http://elijah.poetry.com.ua/texts/google.html>

Обычно поисковые машины используются в графическом режиме. Но их можно запустить и в командном режиме, предусматривая сохранение результатов поиска в файле.

При составлении поискового предписания необходимо обратить внимание на то, как это делается самой поисковой системой. Например, в командной строке google при запросе информации по слову «монография» появляется машинный запрос:

http://www.google.ru/#hl=ru&explds=17259,27555,27744,27789,27868,27937&xhr=t&q=%D0%BC%D0%BE%D0%BD%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F&cp=5&pf=p&sclient=psy&newwindow=1&site=&source=hp&aq=0&aqi=&aql=&oq=%D0%BC%D0%BE%D0%BD%D0%BE%D0%B3&gs_rfai=&pbx=1&fp=1 b3b2485f7ac20fb. В запросе знак % является разделителем символов, а символы представлены в коде ASCII. Знак & разделяет параметры, передаваемые поисковой машине.

В поисковой системе Апорт этот же запрос выглядит значительно проще: <http://sm.aport.ru/search?r=%EC%EE%ED%EE%E3%F0%E0%F4%E8%FF&That=std>.

Базовая поисковая техника:

Для поиска требуемой информации составляется поисковое предписание, содержащее характеризующие эту информацию ключевые слова.

Дополнительные правила составления поискового запроса:

- Google не воспринимает более 10 слов для поиска, включая специальный синтаксис.
- Знак (+) в поисковом предписании используется для поиска наиболее общих слов (пробелы между знаком и помечаемым этим знаком словом не допускаются). Часто используемые слова "I", "a", "the", "of" и т.п. игнорируются поисковиком, но можно заставить его их искать если поставить перед ними "+". Например: "Война +и мир"
- Знак (-) в поисковом предписании используется для исключения термина из поиска (пробелы между знаком и помечаемым этим знаком словом не допускаются)
- Для поиска фразы надо поместить её в кавычки «»
- Знак «*» означает любое слово. Google не поддерживает поиска по корням слов (stemming), то есть возможности использования звёздочки (или другого знака маски) вместо букв в искомом слове. Например, moon* в поисковике, поддерживающем маски, найдёт "moonlight," "moonshot," "moonshadow," и т.д. Google же использует звёздочку как заменитель целого слова. Поиск по фразе "three * mice" в Google даст в результате "three blind mice," "three blue mice," "three red mice," и т.д.
- Обойти лимит в 10 слов можно используя звёздочки. Каждая звёздочка заменяет одно слово. Как оказалось, Google просто не считает количество звёздочек в запросе. "do as * say not as * do" quote origin English usage - замена "do as I say not as I do" quote origin English usage

Так как Google является полнотекстовым поисковиком, он индексирует всё содержимое страниц. Дополнительные команды, называемые спец. синтаксисом (операторы для продвинутого поиска) позволяют пользователям Google искать конкретные части web страниц или тип информации. Это позволяет сузить число результатов поиска. Такие операторы имеют следующий синтаксис: «operator:search_term» (в этом выражении также не должно быть никаких пробелов). Примеры этих операторов:

- **site:** инструктирует Google ограничить поиск конкретным web-сайтом (доменом); название сайта (домена) указывается сразу после двоеточия и без пробела. "site:" сужает поиск до одного сайта или домена верхнего уровня. Например:
 - site:loc.gov
 - site:thomas.loc.gov
 - site:edu
 - site:nc.us
- **filetype:** инструкция произвести поиск только в пределах текста указываемого типа файлов. Тип файла указывается после двоеточия (точку перед расширением файла указывать не нужно). "filetype:" ищет среди расширений файлов, а точнее - в файлах с определённым расширением. Например:
 - homeschooling filetype:pdf
 - "leading economic indicators" filetype:ppt
- **link:** производить поиск внутри гиперссылок содержащих поисковый запрос. "link:" возвращает список страниц, имеющих ссылку на заданную. Например, введите "link:www.google.com" и увидите список сайтов имеющих ссылку на Google. Не обязательно вводить ""; Google проигнорирует эту часть текста даже если её ввести. "link:" отлично работает как с "глубокими" адресами, вроде "http://www.raelity.org/apps/blosxom/", так и с верхнеуровневыми URL, такими как "raelity.org".
- **cash:** оператор демонстрирует версию страницы, которая существовала, когда она индексировалась Google. URL страницы указывается сразу после двоеточия. "cache:" ищет копию страницы проиндексированной Google даже если страница уже недоступна по оригинальному URL или её содержимое полностью изменилось. Например:
 - cache:www.yahoo.com
- **intitle:** производит поиск внутри названия документа. "intitle:" ограничивает поиск до заглавий страниц (titles). Вариации, "allintitle:" ищет страницы в заглавии которых находятся все слова поиска. Например:
 - intitle:"george bush"
 - allintitle:"money supply" economics
- **inurl:** искать внутри URL документа. "inurl:" ограничивает поиск до адресов (URL) страниц. Команда хороша для поиска страниц помощи и поиска, так как они имеют довольно стабильную структуру. "allinurl:" вариация, которая ищет все введённые слова в URL. Например:
 - inurl:help
 - allinurl:search help
- **intext:** ведёт поиск только по тексту страниц (хе. игнорирует текст ссылок, URL, и заглавий). Есть вариация "allintext:", но она плохо ладит с другими командами. Например:
 - intext:"yahoo.com"
 - intext:html
- **inanchor:** ищет текст в якорях ссылок на страницах (anchors). Якори ссылок - это текст описания ссылки. Например, во фрагменте кода HTML [a href="http://www.oreilly.com" O'Reilly and Associates] якорем ссылки является "O'Reilly and Associates." Пример:
 - inanchor:"tom peters"
- **related:** находит страницы, похожие на запрашиваемую. Например, поиск "related:google.com" даст множество поисковиков, включая HotBot, Yahoo!, and Northern Light.

Аналогично:

— related:www.yahoo.com

— related:www.cnn.com

— **info**: предоставляет ссылки на более подробную информацию о запрошенном URL.

Информация включает ссылки на кэш URL, список страниц имеющих ссылки на данную, страницы, связанные с данной, страницы, содержащие данный URL. Например:

— info:www.oreilly.com

— info:www.nytimes.com/technology

Такие операторы можно использовать как по отдельности, так и в различных сочетаниях, в том числе - с различными ключевыми словами. Но некоторые из них отлично работают в сочетании друг с другом, некоторые друг другу мешают, а некоторые просто друг с другом не работают.

Индивидуальные, не сочетающиеся с другими, команды: rphonebook:, bphonebook:, phonebook:, link:. Остальные спец. команды можно смешивать как угодно.

Карта сайта

Чтобы выявить каждую страницу на сайте, Google сканирует его, используя оператор "site:" и дополнительные ключевые слова, которые должны содержаться на каждой странице сайта.

Например, составим запрос такого вида:

```
site:http://www.microsoft.com microsoft.
```

Этот запрос выполняет поиск по слову «microsoft» в пределах сайта http://www.microsoft.com.

Как много страниц на сервере Microsoft содержат слово «Microsoft»? Для выяснения этого вопроса надо иметь в виду, что Google исследует не только содержание страниц, но также их название и URL. Слово «Microsoft» стоит в URL каждой страницы http://www.microsoft.com.

Таким образом - единственным запросом можно инициировать обработку каждой страницы на сайте Microsoft, проиндексированной Google.

Нахождение листинга директории

Листинг директории представляет собой список файлов и директорий удаленного сервера в окне браузера. Такие листинги открывают широкие возможности для углубленного сбора информации. Как правило: такие страницы директорий имеют в Title и теле страницы выражение «Index Of». Отсюда очевидно и строение запроса для поиска таких листингов - это «intitle:index.of». В результате такого запроса будут найдены страницы со словом «index of» в разделе Title документа.

К сожалению - этот запрос вернет слишком большое число страниц не по теме, к примеру, страницы вида:

— Index of Native American Resources on the Internet

— LibDex—Worldwide index of library catalogues

— Iowa State Entomology Index of Internet Resources

Исходя из названий найденных документов, очевидно, что эти страницы не соответствуют

заданному запросу и вряд ли окажутся искомыми списками директорий.

Следующие запросы обеспечат более точные результаты:

```
intitle:index.of "parent directory"  
intitle:index.of name size
```

Такие запросы более точно выдадут то, что нам нужно, поскольку ориентированы не только на фразу «index of» в Title страницы, но и на ключевые слова, всегда имеющиеся в листингах директорий: «parent directory», «name», «size».

Определение версии WEB-сервера

Точная версия программного обеспечения web сервера - это один из элементов, необходимых администратору для точной настройки Web-сайта. Если непосредственно соединиться с сервером, то HTTP (web) заголовки (headers) этого сервера предоставят нужную информацию. Однако можно получить эту информацию *из кэша Google безо всякого соединения с сервером*. Такой метод основан на использовании списка директорий.

Список файлов директории включает имя серверного софта и его версию.

Выглядит такой запрос просто: <<intitle:index.of server.at» Он основан на содержании фразы «index of» в разделе title страницы директории и фразы «server.at», содержащейся в конце любого листинга директории. К примеру, так выглядит запрос, определяющий версию сервера aol.com:

```
«intitle:index.of server.atsite:aol.com».
```

Использование Google в качестве сканера CGI директорий

Для выполнения подобной задачи, CGI сканер изначально знает - какие именно директории нужно искать на сервере. Как правило -это директории, в которых располагаются файлы данных и имеют вид, подобный представленным ниже:

```
/cgi-bin/cgiemail/uargg.txt  
/random_banner/index.cgi  
/random_banner/index.cgi  
/cgi-bin/mailview.cgi  
/cgi-bin/maillist.cgi  
/cgi-bin/userreg.cgi  
/iissamples/ISSamples/SQLQHit.asp  
/iissamples/ISSamples/SQLQHit.asp  
/SiteServer/admin/findvserver.asp  
/scripts/cphost.dll  
/cgi-bin/finger.cgi
```

Зная синтаксис требуемых директорий, а также владея техникой поиска, изложенной выше, можно использовать Google как CGI сканер.

Например, поиск в Google следующего вида: `allinurl:/random_banner/index.cgi` вернет документы с адресами страниц конкретных программ генерации рекламных баннеров.

Использование Google как внутренней поисковой системы Web-сайта.

При поиске информации на серверах можно использовать не только их "родные" формы поиска, но и Google. Например, поисковый запрос: `"george bush" site:nytimes.com` - поиск статей про Дж.Буша на сайте Нью Йорк Таймс.

Исследование Web-страницы с помощью специально составленной программы

Для определения количества форм, гиперссылок, таблиц и структуры таблиц на web-странице разработана программа `index.htm`.

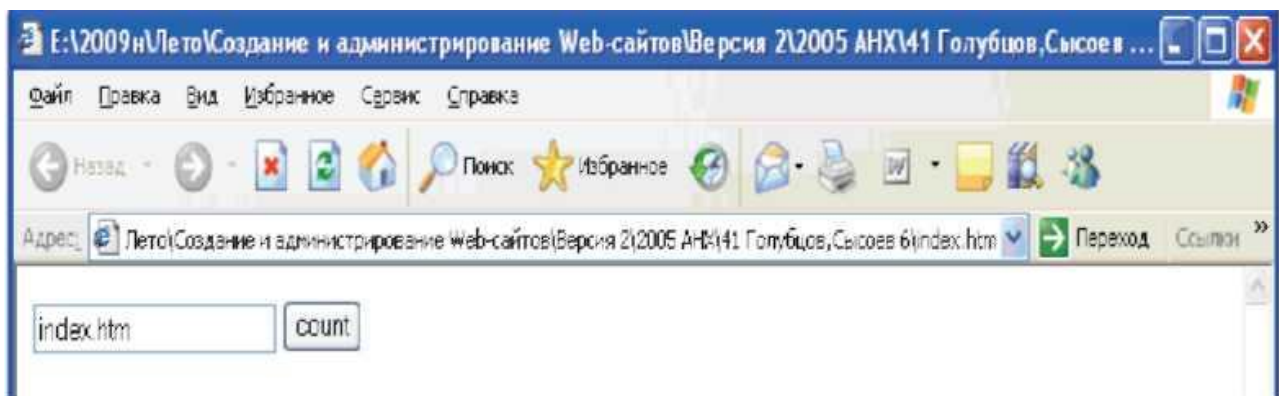
Текст программы `Index.htm`:

```
<html>
<head>
</head>
<script>
function f() {
per=document.fo.tx.value;
a=window.open(per);
i="\nОопМ:"+a.document.all.tags("form").length;
i+="\nСсылк:"+a.document.all.tags("a").length;
i+="\nТаблиц:"+a.document.all.tags("table").length;
i+="\ntr:"+a.document.all.tags("tr").length;
i+="\ntd:"+a.document.all.tags("td").length;
i+="\nbr:"+a.document.all.tags("br").length;
a.alert(i);
}
</script>
<body>
<form name=fo>
<input name="tx" type="text" value="b.htm">
<input name="go" type="button" value="count" onClick="f()">
</form>
</body>
</html>
```

Результат анализа самой себя - т. е. программы `index.htm`:



Поскольку внешний вид окна программы



из него видно, что в данной Web-странице форма - одна, таблиц и ссылок нет.

Исследуем программу 2.htm. Текст программы:

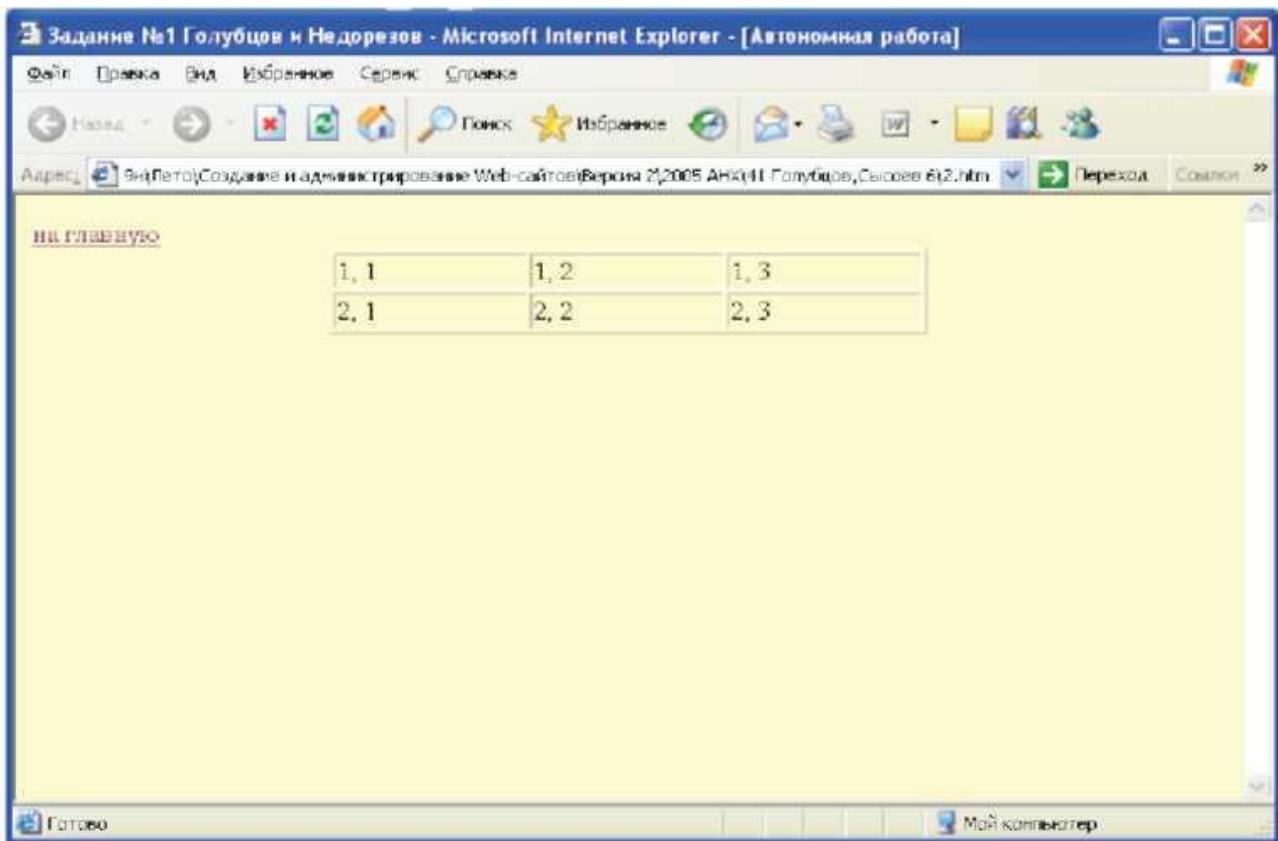
```
<html> <head>
<title>адание №1 Программа, содержащая одну ссылку и одну таблицу</title>
</head>
<body bgcolor="#FFFACD">
<a href="main.htm" alt="на главную">на главную</a> <TABLE WIDTH="50%" ALIGN="center"
BORDER="1">
<TR>
<TD>1, 1</TD>
<TD>1, 2</TD>
<TD>1, 3</TD>
</TR> <TR>
<TD>2, 1</TD>
<TD>2, 2</TD>
```

```
<TD>2, 3</TD>  
</TR>  
</TABLE>  
</body>  
</html>
```

Для неё получаем:



Внешний вид окна программы:



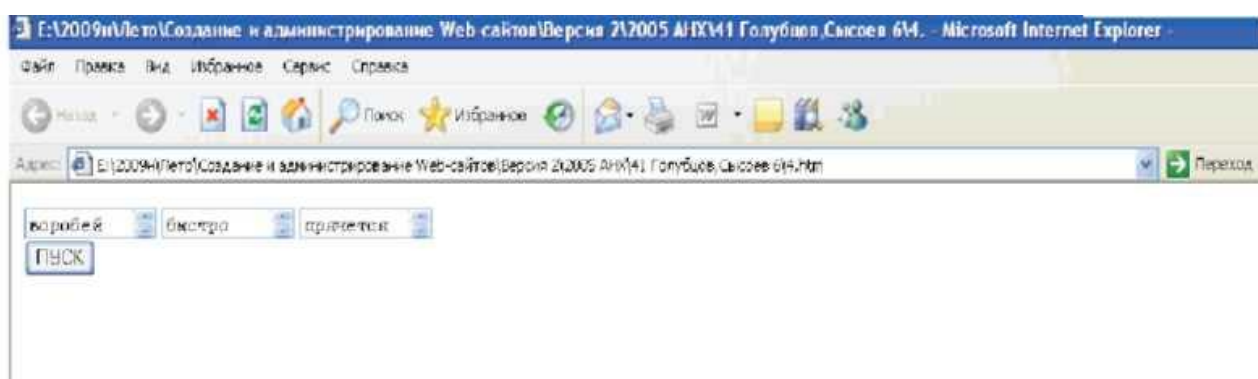
Видно, что в программе есть одна ссылка и одна таблица.

Исследование программы 3.htm. Текст программы:

```
<html> <head>
<title>Псевдоинтеллектуальная программа</title> </head>
<body>
<script language="JavaScript"> function f()
{
var a var b var c
a=Math.floor(Math.random()*(10))+1;
anim = new Array();
anim[1]="акула";
anim[2]="воробей";
anim[3]="заяц";
anim[4]="кузнечик";
anim[5]="лягушка";
anim[6]="попугай";
anim[7]="слон";
anim[8]="собака";
anim[9]="стрекоза";
anim[10]="черепаха";
document.f1.first.value=anim[a];
b=Math.floor(Math.random()*(10))+1;
beta = new Array();
beta[2]="высоко";
beta[3]="громко";
```

```
beta[4]="медленно";
beta[5]="низко";
beta[6]="плохо";
beta[7]="сильно";
beta[8]="слабо";
beta[9]="тихо";
beta[10]="хорошо";
document.f1.second.value=beta[b];
c=Math.floor(Math.random()*(6))+1;
gamma = new Array();
gamma[1]="бегает";
gamma[2]="летает";
gamma[3]="плавает";
gamma[4]="поёт";
gamma[5]="прыгает";
gamma[6]="прячется";
document.f1.third.value=gamma[c];
}
</script>
<form name="f1"><textarea readonly="readonly" cols="10" rows="1" name="first"
aligment="center"></textarea>
<textarea readonly="readonly" cols="10" rows="1" name="second" aligment="center"></textarea>
<textarea readonly="readonly" cols="10" rows="1" name="third" aligment="center"></textarea>
<br>
<input value="ПУСК" name="b1" onclick="f()" type="button">
</form>
</body>
</html>
```

Эта программа представляет собой разновидность псевдоинтеллектуальной программы - собеседника. При нажатии на кнопку Пуск в трёх окнах формы появляются новые слова, выбираемые из заданных трёх наборов случайным образом:



Для программы 4.htm получаем результат исследования:



КОНВЕЙЕРИЗАЦИЯ КОМАНД WINDOWS

Кириченко Александр Аполлонович

Конвейер команд

Выделять из перехваченных данных необходимые для дальнейшего использования можно, используя конвейер команд.

Рассмотрим выдачу, полученную командой `dir` для корневого директория диска СЛ.

```
28.03.2006 23:31 <DIR> 777
25.01.2007 19:16 <DIR> Из HSE
17.09.2005 15:56 <DIR> 2
10.09.2005 16:05 <DIR> 1
21.11.2005 22:47 <DIR> PerfLogs
04.12.2005 19:24 <DIR> temp
15.01.2006 16:57 <DIR> 2006н 02.03.2006 18:12 <DIR> Magic Waterfall Screensaver
04.03.2006 13:52 <DIR> Test_System
02.05.2006 16:22 <DIR> Webservers
02.02.2004 22:26 <DIR> DVD 17.05.2006 15:54 410 543 braun_dyen_kod_da_vinchi.rt f.zip
11.05.2006 16:19 184 320 Olympus 310.doc
06.10.2006 11:12 26 624 bakers word.doc
03.01.2005 16:44 <DIR> lc
25.03.2004 11:09 <DIR> ACTIVstudio
21.02.2005 18:30 182 18.07.2005 13:07 <DIR> 2005-2006 уч год 11.08.2005 11:48 <DIR> JPG-BMP
9 файлов 622 331 байт
29 папок 254 394 368 байт свободно
C:\>_
```

Надо отметить, что этот директорий очень напряжённый и насыщенный. Чаще всего он по команде `dir` не умещается в окне консоли команд.

Здесь приведен лишь фрагмент вывода по команде `dir`. Обратим внимание, что кроме огромного перечня файлов и папок команда `dir` даёт и сводные результаты: в директории хранится 9 файлов общим объёмом 622 331 байт, и 29 папок.

При этом на диске, в корневой директорий которого мы заглядывали, свободно 254 394 368 байт.

Это значит, что команда `dir` даёт достаточное количество информации, чтобы ответить на такие вопросы:

- Сколько файлов содержится в данном каталоге?
- Какой объём занимают файлы, хранящиеся в данном каталоге?
- Сколько папок содержится в данном каталоге?
- Сколько свободного места есть на данном диске?

Но чтобы получить ответы на эти вопросы, надо перевернуть груды информации и найти в

ней то, что Вас интересует.

Ответ должен быть коротким, понятным и информативным.

Чтобы удовлетворить этим требованиям, одной команды `dir` недостаточно.

Возможности этой (и других команд) можно расширить с помощью конвейера.

Вспомним, что есть ещё такая команда `find`, по которой можно среди большого количества строк найти ту, которая больше всего интересует.

Конвейер команд позволяет выдачу, полученную по одной команде, предоставить для анализа другой команде.

Построим конвейер из двух команд: `dir` и `find`.

Первая команда получит информацию о корневом директории диска `C:`, а вторая позволит выделить из этой информации (иногда - очень большого объёма) нужную.

Такое сочетание команд в конвейере позволяет ответить на вопрос:

«Сколько файлов хранится в данной директории и каков их объём?».

Выполним в консоли команд:

```
C:\>dir | find "файлов"  
19 файлов 136 257 байт
```

Здесь для команды `find` в кавычках мы указали поисковое предписание: что искать? А если вместо слова «файлов» указать слово «байт»?

```
C:\>dir | find "байт"  
19 файлов 136 257 байт
```

Получили то же самое.

Внутри одной строки программа `find` неспособна разделить информацию.

Использование конвейера команд

Сколько файлов хранится в данной директории и каков их объём?

Для ответа на вопрос, заданный в заголовке, составим запрос:

```
C:\77>dir | find "файлов"  
19 файлов 136 257 байт
```

Для этой цели можно воспользоваться и другой программой: `findstr`.

```
C:\>dir AX1 | findstr "файлов"  
25 файлов 3 444 924 байт
```

Какие текстовые файлы хранятся в этой директории?

Для ответа на этот вопрос составим запрос:

```
C:\77>dir | find ".txt" и получим ответ:  
05.11.2006 12:36 23 004 test0-1.txt  
04.11.2006 17:38 16 572 test0-1v2.txt
```

В директории найдено два файла с расширением txt.

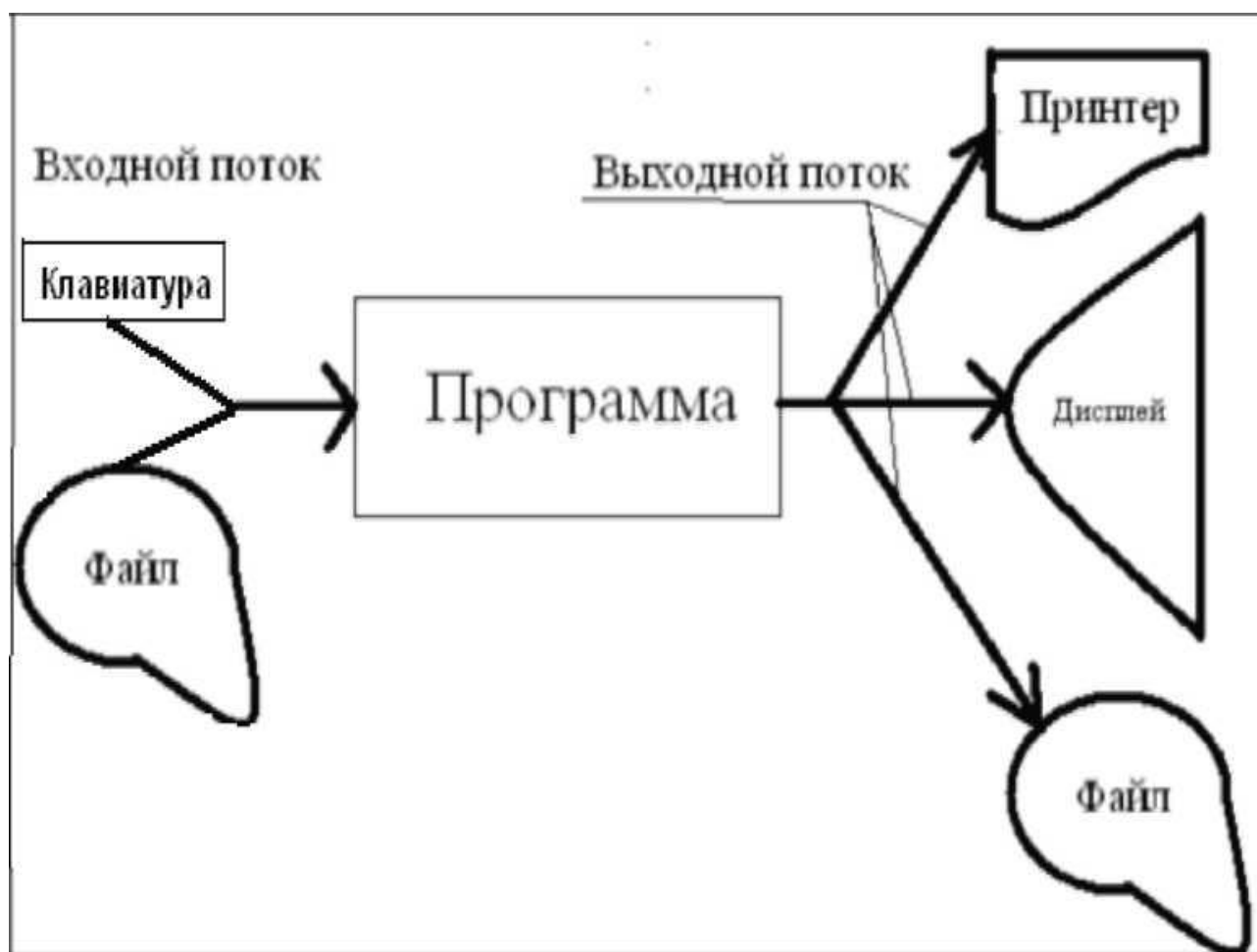
ПРАКТИЧЕСКОЕ ИСПОЛЬЗОВАНИЕ СИСТЕМЫ КОМАНД WINDOWS

Кириченко Александр Аполлонович

Перехват потока

«Перехватить поток» - это значит перенаправить его туда, где он необходим.

Практически любая программа работает по следующей схеме:



Основным входным потоком является поток с клавиатуры. Основным выходным потоком является вывод на дисплей.

Перенаправление потока в операционной системе Windows выполняется с помощью трёх команд:

- <
- >
- >>

Команда "<" - это команда перенаправления **входного потока**.

Острие команды показывает, куда она направлена. Например: **команда**

```
debug.exe < file.txt
```

направляет входной поток из файла file.txt на вход программы debug.exe.

Это значит, что в файле file.txt записана последовательность команд, которую должна выполнить программа debug.exe.

Команда ">" это команда перенаправления выходного потока. Острие команды так же показывает, куда она направлена. Например:

```
Debug.exe > file1.txt.
```

означает, что результаты работы программы debug.exe будут направляться не на экран (как обычно), а в файл file1.txt.

Если этого файла не существует, он будет создан. А если он существует, его содержимое будет заменено результатами работы программы debug.exe. Т. е. информация в этот файл поступит, начиная с самого его начала.

Команда ">>" это команда перенаправления выходного потока. Острие команды так же показывает, куда она направлена.

Например:

```
Debug.exe >> file2.txt.
```

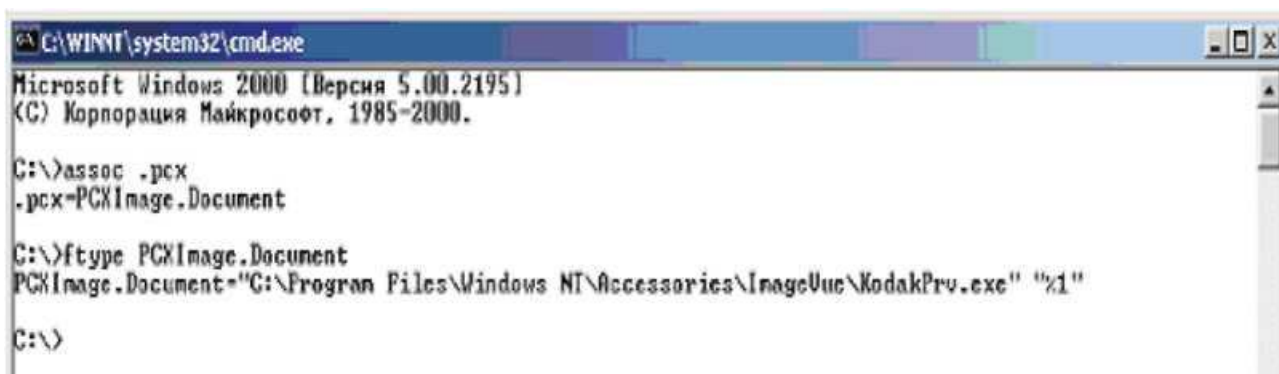
означает, что результаты работы программы debug.exe будут направляться не на экран, а в файл file2.txt.

Если этого файла не существует, он будет создан. А если он существует, его содержимое будет дополнено результатами работы программы debug.exe. Причём эти результаты ничего не уничтожат в файле, они дополнят старое содержимое файла (новые данные будут дописаны в его конец).

Как узнать адрес программы, имеющейся в ОС Windows?

Тип файла характеризуется его расширением. Для обработки файлов определённого типа предназначена своя программа.

- Определить тип файла можно с помощью команды assoc.
- Имя обрабатывающей файлы этого типа программы и её адрес можно получить командой ftype:



```
C:\WINNT\system32\cmd.exe
Microsoft Windows 2000 [Версия 5.00.2195]
(C) Корпорация Майкрософт, 1985-2000.

C:\>assoc .psx
.psx=PCXImage.Document

C:\>ftype PCXImage.Document
PCXImage.Document="C:\Program Files\Windows NT\Accessories\ImageVue\KodakPrv.exe" "%1"

C:\>
```

Пример 1:

Определить, какая программа обрабатывает файлы с расширением .psx?

По команде assoc получаем:

```
.psx=PCXImage.Document
```

В этом ответе находим, что файлы psx относятся к типу «PCXImage.Document».

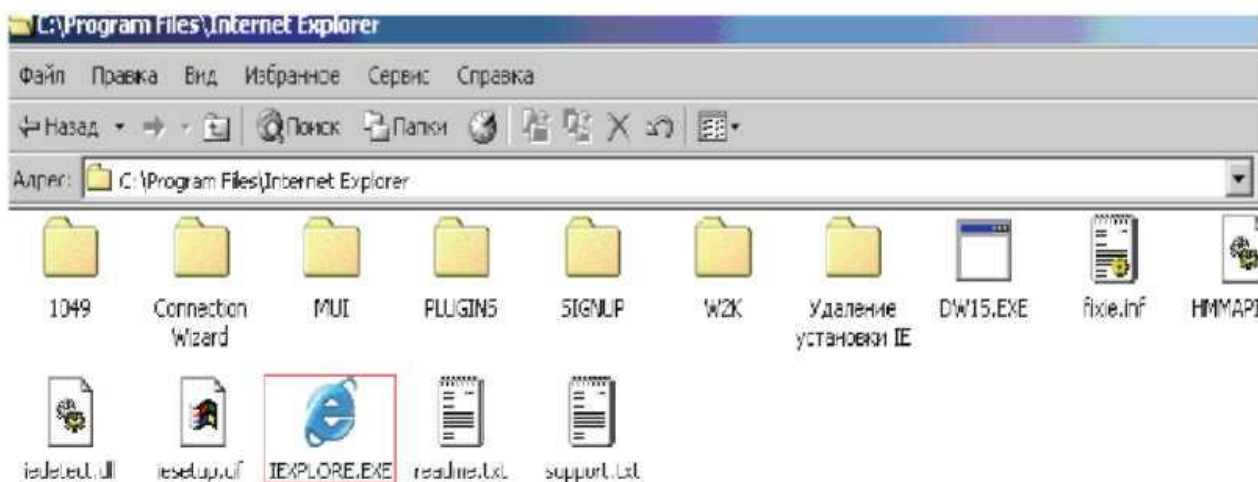
Определим, кто является обработчиком этого типа файлов и где находится сам обработчик:

```
C:\>ftype PCXImage.Document
PCXImage.Document="C:\Program Files\Windows NT\Accessories\ImageVue\KodakPrv.exe"
```

Здесь жирным шрифтом выделен путь к обработчику, курсивом - сам обработчик. После имени обработчика указывается передаваемый обработчику параметр.

Пример 2: надо узнать, где находится обработчик файлов, имеющих расширение htm. Выполняем:

```
C:\>assoc .htm .htm=htmlfile
C:\>ftype htmlfile
htmlfile="C:\Program Files\Internet Explorer\iexplore.exe"
```



Составление отчёта с помощью cmd-файла

Для начала представим себе, как может выглядеть такой отчёт:

Отчет о выполнении пакетного файла 02.cmd Подготовил: студент 1 курса 11 группы:

Ступков А.

создан: 24.04.2009

в

14:39

1) Связать имя папки с виртуальным логическим диском

1. Виртуальный диск O: будет связан с папкой

2. На диске C: будет создана папка work

3. Все файлы маски из будут скопированы в папку

в папке ...

найжены следующие текстовые файлы:

01.cmd

02.cmd

02rep.txt

02rp.cmd

03.cmd

ff.txt

file1.txt

log3.txt

копирование C:_osso\ —> ... файлы скопированы успешно.

теперь папка work содержит следующие текстовые файлы:

01.cmd

02.cmd

02rep.txt

02rp.cmd

03.cmd

ff.txt

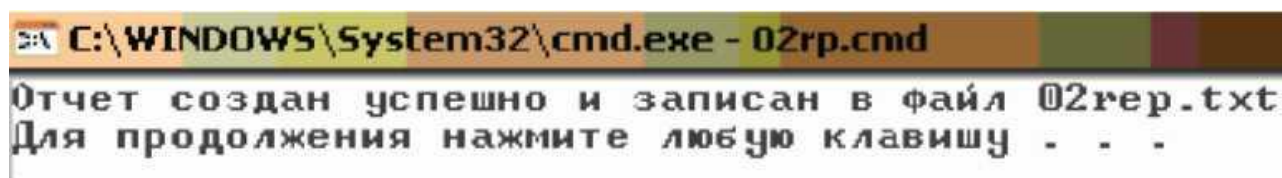
file1.txt

log3.txt

программа завершена.

Несмотря на краткость постановки задачи («Создать файл отчёта»), на самом деле данное задание имеет сложную структуру и включает в себя большое количество задач: для создания отчета надо не просто переписать содержимое экрана в файл, а написать сопроводительный текст, выделить для файла директорию, просмотреть содержимое папки, используемой при создании отчета, вывести содержимое экрана в файл отчета, и др.

Все эти задачи программа выполняет скрытно от пользователя. При запуске данной программы на экране появляется всего лишь информация о расположении файла отчета:



В текстовом редакторе данный командный файл может выглядеть следующим образом:

```
@echo off
echo Отчет о выполнении пакетного файла 02.cmd > 02rep.txt echo. >>
02rep.txt
echo Подготовил: студент 1 курса 11 группы:>> 02rep.txt
echo. >>
02rep.txt
echo Ступков А.>> 02rep.txt
echo. >>
02rep.txt
echo создан: >>
02rep.txt
date/t >>
02rep.txt
echo в >>
02rep.txt
time /t >>
02rep.txt
echo. >>
02rep.txt
echo 1) Связать имя папки с виртуальным логическим диском >>
02rep.txt
echo.
>> 02rep.txt
echo 1. Виртуальный диск O: будет связан с папкой C:\_osso
>> 02rep.txt
echo.
>> 02rep.txt
if not exist c:\_osso\
nul ( echo папка _osso не найдена! >> 02rep.txt
goto END)
if exist O:\nul ( echo Создание диска отменено: диск O: уже существует! >> 02rep.txt goto NOSU
)
else ( echo создание виртуального диска O: ... >> 02rep.txt
)
SUBST O: C:\_osso
if errorlevel 1 (echo не удалось создать диск O:! >> 02rep.txt goto END)
else ( echo диск O: создан и сопоставлен пути C:\_osso успешно.>> 02rep.txt)
:NOSU
echo. >> 02rep.txt
echo 2. На диске C: будет создана папка work >> 02rep.txt echo. >> 02rep.txt
if exist C:\work\nul ( echo Создание папки отменено: папка work уже существует! >> 02rep.txt
goto AFMD )
else ( echo создание папки c:\work ... >> 02rep.txt)
```

```
MD C:\work
if errorlevel 1 ( echo не удалось создать папку work! >>
02rep.txt goto END )
else ( echo папка work создана успешно. >> 02rep.txt) :AFMD
echo. >> 02rep.txt
echo 3. Все файлы %1 маски из C:\_osso >> 02rep.txt
echo будут скопированы в папку C:\work >> 02rep.txt
echo. >> 02rep.txt
echo поиск текстовых файлов в папке c:\_osso... >> 02rep.txt if not exist C:\_osso\%1 (
echo текстовые файлы не найдены. >> 02rep.txt
goto END)
echo. >> 02rep.txt
echo найдены следующие текстовые файлы: >> 02rep.txt
echo _____ >> 02rep.txt
echo /\ >> 02rep.txt
DIR C:\_osso\%1 /B /P >> 02rep.txt
echo \_____ / >> 02rep.txt
echo. >> 02rep.txt
echo копирование C:\_osso\%1 —Л> C:\work ... >> 02rep.txt
echo. >> 02rep.txt
COPY C:\_osso\%1 C:\workecho. >> 02rep.txt
if errorlevel 1 ( echo возникли ошибки... >> 02rep.txt
goto END )
else ( echo файлы копированы успешно. >> 02rep.txt)
echo. >>
02rep.txt
echo теперь папка work содержит следующие текстовые файлы: >> 02rep.txt
echo _ >>
02rep.txt
_____
echo / \ >>
02rep.txt
DIR C:\work\%1 /B /P >>
02rep.txt
echo \_____ / >>
02rep.txt :END
echo. >>
02rep.txt
echo программа завершена. >>
02rep.txt
echo.cls
echo Отчет создан успешно и записан в файл 02rep.txt
pause
cls
```

Создание меню

Меню в командной строке Windows2000/XP создаётся из нескольких BAT- или CMD- файлов, находящихся в одной директории.

При создании меню создается cmd-файл, содержащий пункты меню и команду PROMPT, которая содержит текст «Введите номер выбранного пункта и нажмите ENTER...».

Кроме этого файла создаются cmd-файлы с именами 1,2,3,4 и т.д., эти файлы выполняют действия пунктов, которые содержатся в меню. Каждый из этих файлов заканчивается командой вызова основного cmd-файла меню.

Последний файл меню содержит команду выхода из меню.

Для входа в меню в командной строке набирается имя файла который содержит пункты меню и команду Prompt. При необходимости командная строка может содержать и замещаемые параметры.

Т.к. для создания меню применяется команда PROMPT, а в примере также используется архиватор WINRAR, в приложении приводятся соответствующие Help.

Пример создания меню с помощью cmd-файлов.

Шаг №1- Создаём файл menu.cmd

```
C:\>copy con menu.cmd
Echo off Cls
Echo Стартовое меню
Echo. Echo.
Echo 1.  Просмотреть  содержимое файла.
Echo. Echo.
Echo 2.  Архивировать  файл и удалить исходный.
Echo. Echo.
Echo 3.  Разархивировать файл и удалить архив.
Echo. Echo.
Echo 4.  Выход.
Echo.
Prompt Введите номер и нажмите ENTER...
```

Команда Prompt в процессе работы командного файла выводит новое приглашение командной строки - надпись «Введите номер и нажмите ENTER...». После этой надписи должна быть набрана цифра 1,2,3 или 4 - эти цифры соответствуют именам cmd-файлов, пример создания которых будет приведён ниже, эти cmd-файлы будут выполнять действия, предлагаемые в меню.

При запуске созданного файла menu из командной строки на экране появится меню:

```
Стартовое меню
1.  Просмотреть содержимое файла
2.  Архивировать файл и удалить исходный
3.  Разархивировать файлы и удалить архив
4.  Выход
Введите номер и нажмите ENTER...
```

Теперь создадим файлы с именами 1,2,3 и 4. Эти файлы будут выполнять следующие действия:

1. Просматривать содержимое введённого замещающим параметром файла.
2. Архивировать указанный файл и удалять исходный файл.
3. Разархивировать указанный архив в указанную замещающим параметром директорию, и удалять архив.
4. Выход из меню.

Структура файла 1.cmd и обращение к нему:

Создадим файл 1.cmd

Введите номер и нажмите ENTER...

```
C:\>copy con 1 .cmd
type %1 |more
pause
menu
```

%1 - замещаемый параметр, т.е. вместо %1 из командной строки вводится путь и имя просматриваемого файла.

|more - этот параметр останавливает содержимое экрана в том случае, если информация о содержимом файла не помещается на экран.

Pause - останавливает экран до нажатия любой клавиши, после чего срабатывает команда menu (имя файла который содержит внешний вид меню) и происходит возврат в меню.

Если команда pause отсутствует, то в случае, если содержимое файла помещается на экране, оно не будет задерживаться на экране, а будет происходить быстрый возврат в меню. Выводимое изображение просто промелькнёт, после чего на экран будет снова вызвано меню.

Пример 1. Обращение к файлу 1.cmd: информация из файла выводится на экран (обращение выделено):

Стартовое меню

1. Просмотреть содержимое файла
2. Архивировать файл и удалить исходный
3. Разархивировать файлы и удалить архив
4. Выход

Введите номер и нажмите ENTER... C:\vns\l\l.txt

Выводиться текст

Pause

Для продолжения нажмите любую клавишу...

После нажатия любой клавиши произойдёт возврат в меню.

Структура файла 2.cmd и обращение к нему:

Создадим файл 2.cmd

```
cd %i
rar a %2 %3
del «3
cd c:\
menu
```

cd %1 - эта команда автоматически переводит нас в папку где установлен архиватор, делая её активной. (Папка с местонахождением архиватора задаётся замещаемым параметром. (Обычно - это files\Winrar))

Rar a %2 %3 - команда архивирует указанный файл (например, если для %3 задано C:\vms\11\t1.txt) в архив tl.rar (если для %2 задано C:\vms\11\tl.rar).

Del %3 - команда удаляет файл который был заархивирован (когда для %3 было задано)

Обращение к файлу при выборе пункта 2 меню будет выглядеть следующим образом (обращение выделено):

Стартовое меню

```
1. Просмотреть содержимое ф аила
2. Архивировать ф айл и удалить исходный
3. Разархивировать ф айлы и удалить архив 4 RhiwnA
Введите номер и нажмите ENTER. C:\Program files\Winrar C:\vms\11\t1.rar
Введите номер и нажмите ENTER. cd c:\program files\Winrar
Введите номер и нажмите ENTER. rar a c:\vms\11\t1.rar c:\vmz\11\t1.txt
RAR 2.71 Авторские права (C) 1993-2000 Евгений Рошал 20 июня 2000
Незарегистрированная копия. Наберите RAR -? для получения справки
Незарегистрированная копия. Пожалуйста, зарегистрируйтесь.
Создание архива c:\vms\11\t1.rar
Добавление vms\11\t1.txt OK
Готово
Введите номер и нажмите ENTER...del c:\vms\11\t1.txt
Введите номер и нажмите ENTER...cd c:\
```

При архивировании файла информация пролетает очень быстро, информация была снята в результате добавления в конец команды pause.

Структура файла 3.cmd и обращение к нему:

Создадим файл 3.cmd

```
cd %2
rar e %2 %3
del %1
cd\
menu
```

Cd %1 - эта команда автоматически переводит нас в папку, где установлен архиватор, делая её активной. (Папка с местонахождением архиватора задаётся замещаемым параметром (C:\Program files\Winrar))

Rar e %2 %3 - команда разархивирует указанный архив (для %2 = (C:\vms\11\t1.rar)). в указанную директорию (для %3 = (C:\vms\11\)).

Del %2 - команда удаляет архив который был разархивирован (для %2 = (C:\vms\11\t1.rar))

Обращение к файлу выглядит следующим образом (обращение выделено):

Стартовое меню

1. Просмотреть содержимое файла
2. Архивировать файл и удалить исходный
3. Разархивировать файлы и удалить архив
4. Выход

Введите номер и нажмите ENTER...C:\Program files\Winrar C:\vms\11\t1.rar

Введите номер и нажмите ENTER...cd c:\program files\Winrar

Введите номер и нажмите ENTER...rar e c:\vms\11\t1.rar c:\vms\11\

Извлечение из c:\vms\11\t1.rar

Извлечение c:\vms\11\t1.txt OK

Введите номер и нажмите ENTER...del c:\vms\11\t1.rar

Введите номер и нажмите ENTER...cd c:\

При разархивировании файла информация пролетает очень быстро, информация была снята в результате добавления в конец команды pause.

Структура файла 4.cmd и обращение к нему:

Создадим файл 4.cmd

Exit

Файл 4.cmd содержит одну команду exit, это команда выхода из меню.

Обращение к файлу:

Стартовое меню

1. Просмотреть содержимое файла
2. Заархивировать файл и удалить его
3. Разархивировать файлы и удалить архив
4. Выход

Введите номер и нажмите ENTER...4

При обращении к файлу 4.cmd происходит выход из меню.

Исследование программы Find.exe

Утилита Find.exe используется для нахождения отрывка текста в одном или нескольких файлах.

1) Создадим текстовый файл Sample.txt на диске c:

Белеет парус одинокий
В тумане моря голубом
Что ищешь ты в стране далекой

2) формат команды :

```
FIND [/V] [/C] [/N] [/I] "строка" [[диск:][путь]имя_файла[ ...]]
```

где "строка" - Искомая строка (шаблон для поиска).

[диск:][путь]имя_файла - Один или несколько файлов, в которых выполняется поиск.

/V - Вывод всех строк, НЕ содержащих заданную строку.

```
C:\>find /v "моря" c:\samp1e.bat  
.....C:\SAMPLE.TXT  
Белеет парус одинокий  
Что ищешь ты в стране далекой
```

/C - Вывод только общего числа строк, содержащих заданную строку.

```
C:\>find /c "моря" c:\sample.txt  
.....C:\SAMPLE.TXT: 1
```

/N - Вывод номеров отображаемых строк.

```
C:\>find /n "моря" c:\sample.txt  
.....C:\SAMPLE.TXT  
[2]В тумане моря голубом
```

/I - Поиск без учета регистра символов.

```
C:\>fmd /i "моря" c:\sample.txt  
.....C:\SAMPLE.TXT  
В тумане моря голубом
```

Создание командного файла, использующего программу Find.exe

Создадим командный файл Fstring.bat, использующий программу Find.exe.

```
C:\>copy con fstring.bat  
find /%1 "%2" C:/%3  
^Z  
Скопировано файлов: 1.
```

В данном примере:

%1 - переменная, запрашивающая параметр вывода ([/V] [/C] [/N] [/I]).

%2 - переменная, запрашивающая строку для поиска.

%3 - переменная, запрашивающая файл с диска "C:" в котором будет искаться данная строка.

Далее создадим текстовый файл в котором мы будем искать строку.

```
I asked him about comandnles  
And what do you think he answered  
He told that it is very simple  
To make it properly
```

Теперь введем команду `fstring 12 3`, где 1, 2 и 3 - параметры функции.

Например:

```
C:\>fstring n do sample.txt  
C:\>find/n "do" c:/sample.txt  
.....C:/SAMPLE.TXT  
[2]And what do you think he answered
```

Управление из командной строки программой msinfo32

Помимо средства «Сведения о системе» для просмотра данных об управляемом компьютере можно использовать команду `msinfo32`.

`Msinfo32` служит для отображения подробных сведений об оборудовании, системных компонентах и среде программного обеспечения.

Синтаксис

```
msinfo32 [/?] [/pch] [/nfo имяфайла] [/report имяфайла] [/computer имя компьютера]  
[/showcategories] [/category код категории] [/categories код категории]
```

Параметры

имяфайла - файл, который требуется открыть. Файл может иметь расширение NFO, XML, TXT или CAB.

/? - отображение справки по команде `msinfo32`.

/pch - отображение журнала.

/nfo имяфайла - сохранение экспортированного файла как NFO-файла.

/report имяфайла - сохранение экспортированного файла как TXT-файла.

/computer имякомпьютера - открытие окна сведений о системе для указанного удаленного компьютера.

/showcategories - открытие окна сведений о системе, содержащего все доступные коды категорий.

/category код категории - Открытие окна сведений о системе, в котором выбрана указанная категория. Для отображения списка доступных кодов категорий служит параметр **/showcategories**.

/categories кодкатегории - открытие окна сведений о системе, содержащего только указанные категории. Вывод также ограничивается только выбранными категориями. Для отображения списка доступных кодов категорий служит параметр **/showcategories**

Примеры

Чтобы получить список доступных кодов категорий, введите:

```
msinfo32 /showcategories
```

Чтобы открыть окно сведений о системе, содержащее все доступные сведения, кроме сведений о загруженных модулях, введите:

```
msinfo32 /categories +all -loadedmodules
```

Чтобы открыть окно сведений о системе и создать NFO-файл `syssum.nfo`, содержащий сведения категории «Сведения о системе», введите:

```
msinfo32 /nfo syssum.nfo /categories +systemsummary
```

Чтобы вывести сведения о конфликте ресурсов и создать NFO-файл `conflicts.nfo`, содержащий сведения о конфликтах ресурсов, введите:

```
msinfo32 /nfo conflicts.nfo /categories  
+componentsproblemdevices+resourcesconflicts+resourcesforcedhardware
```

Команда FINDSTR

Findstr - поиск образцов текста в файлах с использованием регулярных выражений.

Формат команды:

```
findstr [/b] [/e] [/l] [/r] [/s] [/i] [/x] [/v] [/n] [/m] [/o] [/p] [/offline] [/g:файл] [/f:файл] [/c:строка]  
[/d:СписокКаталогов] [/a:АтрибутЦвета] [строки] [[диск:][путь] ИмяФайла [...]]
```

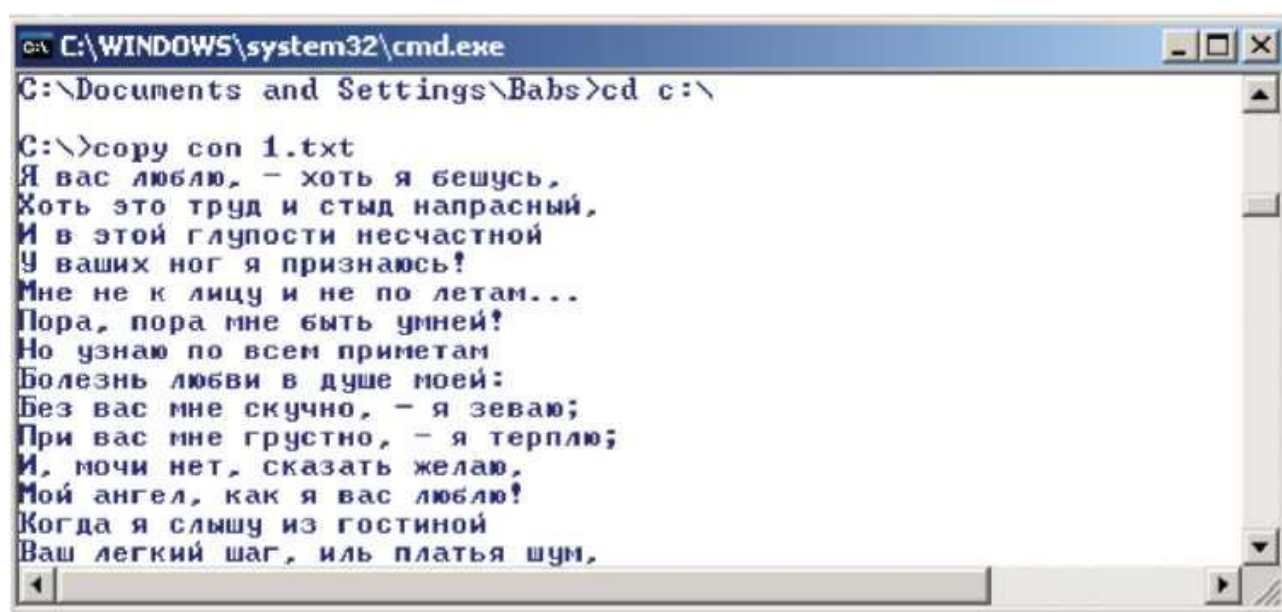
, где строки - текст, поиск которого производится в файле, заданном параметром ИмяФайла [диск:][путь] ИмяФайла [...] - файл или несколько файлов для поиска

Ключи команды:

/b	Сравнивает шаблон с началом строки.
/e	Сравнивает шаблон с концом строки.
/l	Использует заданную строку буквально.
/r	Использует строку поиска как регулярное выражение. Команда Findstr интерпретирует все метасимволы как регулярные выражения, если не используется ключ /l.
/s	Задаёт поиск файлов в текущем каталоге и его подкаталогах.
/i	Задаёт поиск без различия строчных и заглавных букв.
/x	Печатает точно совпавшие строки.
/v	Печатает строки, не содержащие совпадений.
/n	Печатает в начале совпавшей строки ее номер.
/m	Печатает только имя файла при обнаружении совпадения.
/o	Печатает смещение перед выводом строки с совпадением.
/p	Пропускает файлы с непечатаемыми символами.
/offline	Обработка файлов с автономным атрибутом.
/f:файл	Читает список из заданного файла.
/c:строка	Использует заданный текст как литеральную строку поиска.
/g:файл	Получает строки поиска из заданного файла.
/d:Список каталогов	Ищет в списке каталогов, разделенном запятыми.
/a:Атрибут цвета	Задаёт атрибуты цвета двумя шестнадцатеричными цифрами.

Пример использования команды findstr:

Сначала создадим текстовый файл 1.txt, используя команду copy con:



```

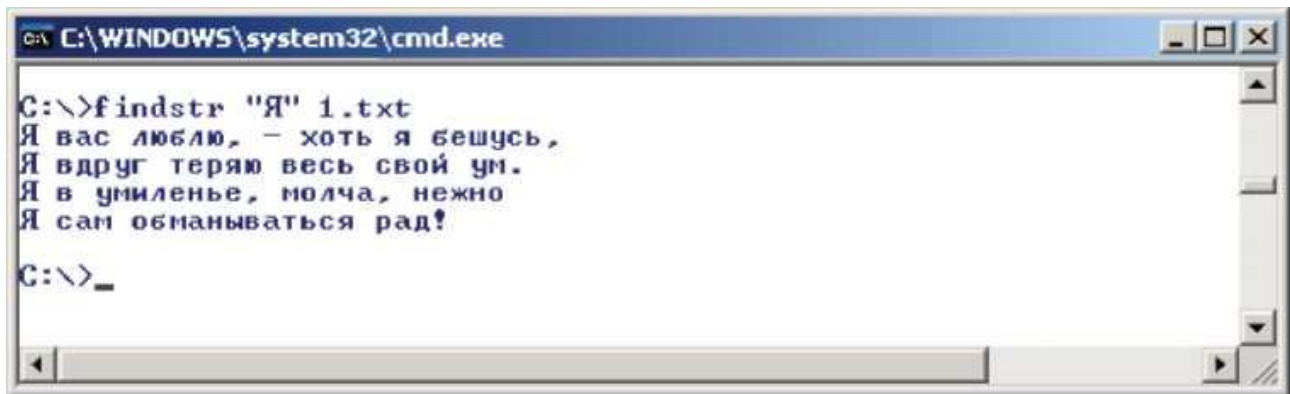
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Babs>cd c:\

C:\>copy con 1.txt
Я вас люблю, - хоть я бешусь,
Хоть это труд и стыд напрасный,
И в этой глупости несчастной
У ваших ног я признаюсь!
Мне не к лицу и не по летам...
Пора, пора мне быть умней!
Но узнаю по всем приметам
Болезнь любви в душе моей:
Без вас мне скучно, - я зеваю;
При вас мне грустно, - я терплю;
И, ночи нет, сказать желаю,
Мой ангел, как я вас люблю!
Когда я слышу из гостиной
Ваш легкий шаг, иль платья шум,

```

Теперь найдем строки содержащие слово «Я». Для этого введем команду:

```
C:\>findstr "Я" 1.txt
```



```
C:\WINDOWS\system32\cmd.exe
C:\>findstr "Я" 1.txt
Я вас люблю, - хоть я бешусь,
Я вдруг теряю весь свой ум.
Я в умиление, молча, нежно
Я сам обманываться рад!
C:\>_
```

Если нам необходимо найти целую строку или определенное словосочетание, то воспользуемся командой:

```
C:\>findstr /c:"обманываться рад" 1.txt
```



```
C:\WINDOWS\system32\cmd.exe
C:\>findstr /c:"обманываться рад" 1.txt
Я сам обманываться рад!
C:\>_
```

Для вывода на экран строк не содержащих совпадений воспользуемся командой:

```
C:\>findstr /v /c:"обманываться рад" 1.txt
```

Команда SORT

Команда SORT считывает входящие данные, сортирует их. Результат может быть выведен на экран, в файл или на другое устройство.

Формат команды:

```
SORT [/R] [/+n] [/M килобайтов] [/I язык] [/REC символов] [[диск1:][путь1]имя_файла1] [/T [диск2:][путь2]] [/O [диск3:][путь3]имя_файла3]
```

, где ключи команды:

/r

Изменяет порядок сортировки, задает сортировку от Z к A, затем от 9 к 0.

/+n

Указывает номер позиции символа, n, начиная с которой sort выполняет сравнение.

/m килобайты

Указывает количество основной памяти, используемой для сортировки (в килобайтах).

/l язык

Переопределяет порядок сортировки символов, задаваемый языком, который используется системой по умолчанию (т. е. языком и страной/регионом, выбранными при установке).

/гес символы

Указывает максимальное число символов в записи (или строке) входного файла. (Значение по умолчанию равно 4096, максимальное значение — 65535).

[диск 1:][путь1]имя файла 1

Указывает сортируемый файл. Если имя не указано, сортируется стандартный ввод. При указании входного файла команда работает быстрее, чем при перенаправлении того же файла на стандартный ввод.

/t [диск2:][путь2]

Указывает каталог для хранения рабочих данных команды sort, если сортируемые данные не помещаются в основной памяти. По умолчанию используется системный временный каталог.

/o [диск3:][путь3]имя файла3

Указывает файл для хранения отсортированного ввода. Если параметр не указан, данные будут записываться в стандартный вывод. При указании выходного файла команда работает быстрее, чем при перенаправлении того же файла на стандартный вывод.

Так же команда SORT без параметров может быть использована для сортировки информации, вводимой с клавиатуры. Затем следует ввести текст, который необходимо отсортировать; каждая строка заканчивается нажатием клавиши ENTER. Когда ввод будет закончен, следует нажать комбинацию CTRL+Z и ENTER. Команда sort выведет на экран введенные строки, отсортированные в алфавитном порядке.

Команда - FTYPЕ

Просмотр и изменение типов файлов, сопоставленных с расширением имен файлов

FTYPЕ [типФайлов [= [команднаяСтрокаОткрытия]]]

типФайлов - Тип файлов для просмотра или изменения

команднаяСтрокаОткрытия - Команда, используемая для открытия файлов указанного типа.

- Команда FTYPЕ без параметров выводит текущий список типов файлов, для которых определены командные строки открытия.
- Если указан только тип файла, FTYPЕ выводит командную строку открытия для этого типа файлов.
- Если после знака равенства не указана строка открытия, FTYPЕ удалит текущее сопоставление

для указанного типа файлов.

- При вызове командной строки переменные %0 и %1 заменяются на имя файла, запускаемого с помощью сопоставления.
- Вместо переменной %* подставляются все параметры, а переменные %2, %3 и т.д. заменяются, соответственно, на первый, второй и другие параметры.
- Вместо переменной %~n подставляются все оставшиеся параметры, начиная с n, где n является числом от 2 до 9. Например:

```
ASSOC .pl=PerlScript
```

```
FTYPE PerlScript=perl.exe %1 %*
```

Эти команды позволят вызывать обработчик команд Perl следующим образом:

```
script.pl 1 2 3
```

- Если желательно избежать постоянного ввода расширения имен файлов, введите следующую команду:

```
set PATHEXT=.pl;%PATHEXT%
```

Теперь обработчик команд вызывается еще проще:

```
script 1 2 3
```

Команда subst

Сопоставление имени диска указанному пути.

Формат команды

```
SUBST [диск1: [диск2:]путь] SUBST диск1: /D
```

где

диск1: Виртуальный диск, который сопоставляется указанному пути.

[диск:]путь Физический диск и путь, которым сопоставляется виртуальный диск.

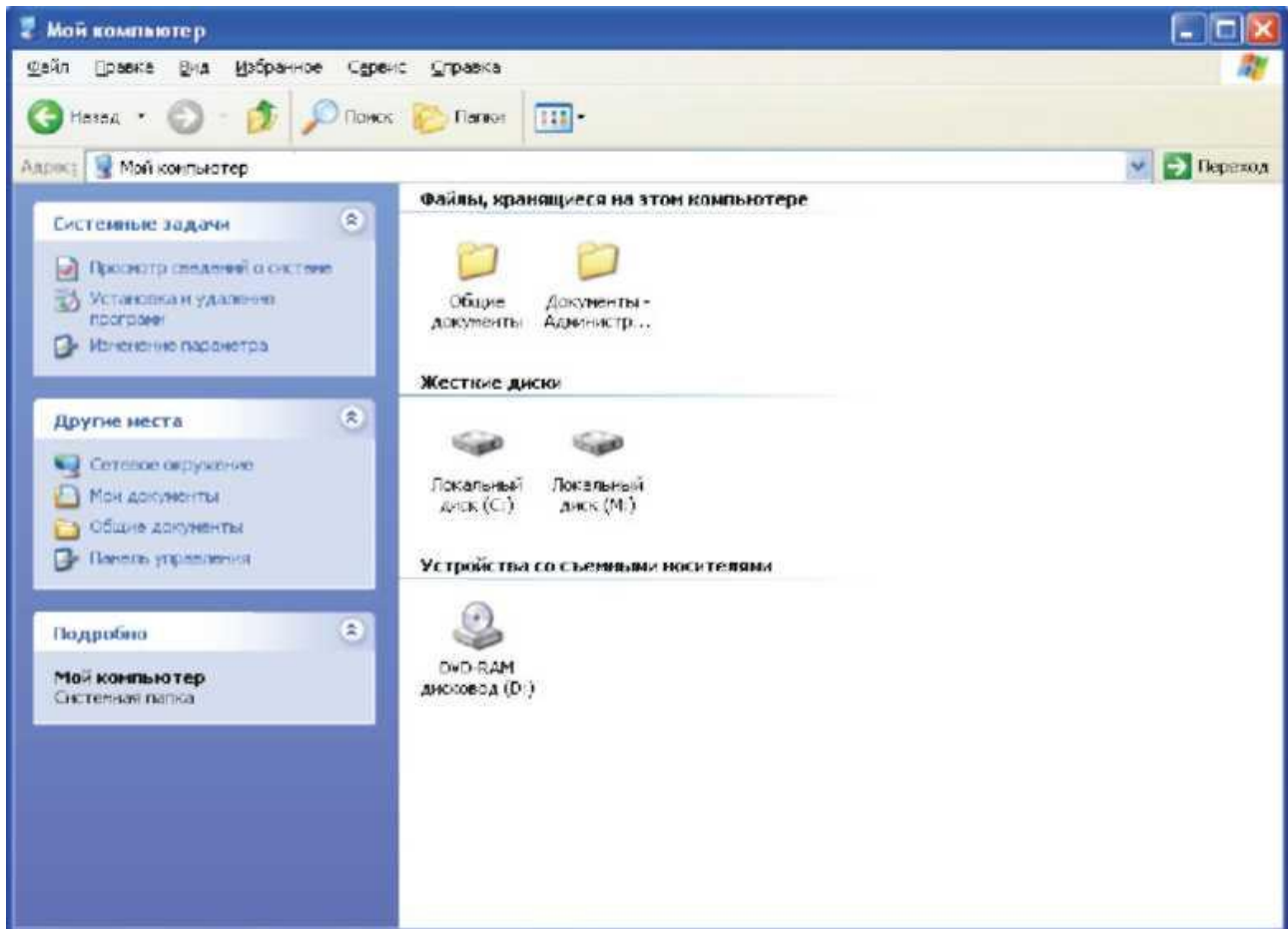
/D Удаление ранее созданного виртуального диска.

SUBST без параметров - вывод текущего списка виртуальных дисков.

Создадим виртуальный диск M, который сопоставляется с путём :

```
C:\WINXP\system32\cmd.exe
C:\Documents and Settings\Администратор>subst m: c:\neudisk
C:\Documents and Settings\Администратор>_
```

Таким образом, в папке «Мой компьютер» появится новый диск M:



Теперь удалим этот диск:

```
C:\WINXP\system32\cmd.exe
C:\Documents and Settings\Администратор>subst n: /d
C:\Documents and Settings\Администратор>_
```

Соответственно, он перестанет существовать и в «Моём компьютере».

Исправление тестирующей программы за счёт

комплексирования программных средств

Постановка задачи: имеется программа для тестирования знаний по методу «один правильный ответ». Программа может быть использована для тестирования разнообразных знаний. С этой целью задаваемые вопросы, возможные ответы на них и правильные ответы содержатся в текстовом файле с расширением .dat ,

Необходимо скрыть текстовый файл от тестируемых.

При решении этой задачи реализован следующий подход: на языке C# составлена программа зашифровки текста (shifruem.cs). С её помощью текст превращается в зашифрованный файл с расширением ".s".

Во время тестирования учащимся предоставляются:

- Текстовый файл с расширением ".s"
- Программа расшифровки "d.exe", создающая файл с расширением .dat
- Тестирующая программа tt.exe
- Два cmd-файла: "begin.cmd" и "tt1.cmd".

Файл "begin.cmd" создаёт на диске C: скрытую папку, путь к которой превращает в логический диск W. На этот диск переписываются тестирующая программа и восстановленный текстовый файл с вопросами теста и ответами на них. Правильный ответ располагается всегда первым. Логический диск W существует только во время тестирования и удаляется при выходе из программы.

Сказать, что так создана идеальная система защиты от несанкционированного просмотра ключевого файла, нельзя. Но для того, чтобы найти этот файл, надо понимать команды, в cmd-файлах. А это требует наличия у тестируемого определённой квалификации.

Естественно, воспользоваться таким способом тестирования можно только однажды.

Далее приводятся: содержание подготовленной к тестированию папки, тексты cmd-файлов, внешний вид экрана во время тестирования и текст программы shifruem.cs.

Подготовленная к тестированию папка содержит:



Текст файла Begin.cmd:

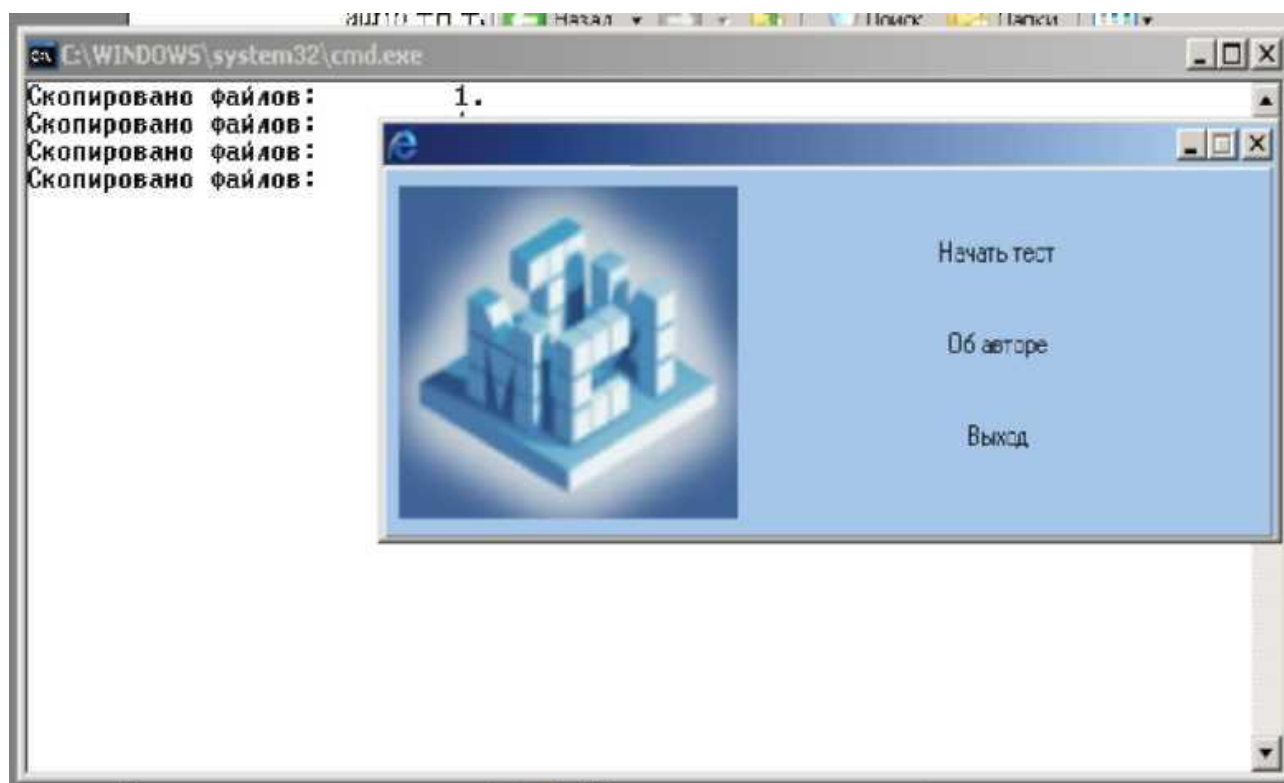
```
@echo off color F0
if not exist c:\12 md c:\12\
if not exist c:\12\weojrdkci md c:\12\weojrdkci
if not exist w: subst w: c:\12\weojrdkci
attrib +H +S /S /D c:\12\weojrdkci D.exe
copy test1.dat w:\test1.dat
copy 2.s w:\2.s
copy Tt.exe w:\Tt.exe
copy tt1.cmd w:\tt1.cmd
del test1.dat
rem del D.exe 2.s Tt.exe tt1.cmd test1.dat w:
tt1.cmd
```

Текст файла tt1.cmd:

```
@echo off
tt.exe
c:
subst w: /d
```

Использование такого комплекса программ приводит к тому, что расшифрованного файла с тестом не видно - он спрятан.

В данном случае цель достигается за счёт комплексирования программных средств: не имеющего исходных текстов файла тестера (не только не имеющего исходных текстов, но и неизвестно, на каком языке написанного), двух простых программ -написанных на языке C#, и двух командных файлов операционной системы Windows.



А зашифровка ведётся программой Shifruem.exe.

Текст программы Shifruem.exe:

```
using System; using System.IO; using System.Text; namespace FileStream {
class Class1 {
[STAThread]
static void Main(string[] args) {
//Ооздаем экземпляр Chitat класса StreamReader и указываем путь к файлу и его название
StreamReader Chitat = new StreamReader("test1.txt",System.Text.Encoding.GetEncoding("w indows-
1251"));
// Объявляем переменную A.
string A;
//В переменную A считываем содержимое файла.
A = Chitat.ReadToEnd();
//Закрываем поток
Chitat.Close();
//объявляем переменную temp, в которой накапливается //зашифрованный текст
string temp = "";
// Шифруем!!!
for (int i = 0; i < A.Length; i++) temp += (char)(A[i] + 1);
// Выводим на экран : //string AC=" Вот , что сделали : ";
//Console.WriteLine(AC);
//Console.WriteLine(temp); //Останавливаемся для просмотра: //Console.ReadLine();
//Создаем экземпляр Pisat
//класса StreamWriter StreamWriter Pisat = new StreamWriter("2.s", true,
System.Text.Encoding.Unicode);
//Вызываем метод Write экземпляра Pisat Pisat.Write(temp); //Закрываем поток
Pisat.Close();
```

}
}
}

КЛАССЫ .NET ДЛЯ РАБОТЫ С ПРОЦЕССАМИ

Кириченко Александр Аполлонович

Пространство имён System.Diagnostics содержит 7 классов и 2 перечисления.

Для управления внешними (по отношению к Си-программам) программами основными являются 2 класса: Process и ProcessStartInfo.

Класс Process предоставляет доступ к локальным и удалённым процессам и позволяет запускать и останавливать локальные системные процессы.

В MSDN каждый класс характеризуется по типовой схеме: общее описание класса, перечисление членов класса (конструктор, методы, свойства, события), подробное описание членов класса с примерами их использования. Такая же схема выдерживается и при описании классов Process и ProcessStartInfo.

Электронной вычислительной машине для выполнения любой программы нужны определённые ресурсы: память, устройства, файлы, адресное пространство, процессорное время, и др. Программа, которой операционная система выделила необходимые для работы ресурсы, образует процесс, а точнее - процесс операционной системы, или системный процесс.

Последовательность создания процесса в C#-программе предусматривает создание экземпляра класса (компонента Process) и связывание этого экземпляра с системным процессом (т.е. процессом ОС).

Создание экземпляра производится конструктором класса. При этом экземпляру (компоненту, или созданному объекту) присваивается имя. Если процесс создаётся в классе MyProcess и ему присваивается имя process1, то делается это с использованием оператора new:

```
MyProcess process1 = new MyProcess();
```

Связывание создаваемого процесса с системным производится методом Start().

При создании процесса обязательно должно быть определено значение свойства StartInfo.FileName. Обычно это свойство определяется перед связыванием с системным процессом (т. е. перед командой Start()) и определяет ресурс процесса - используемую программу или документ, обрабатываемый этой программой.

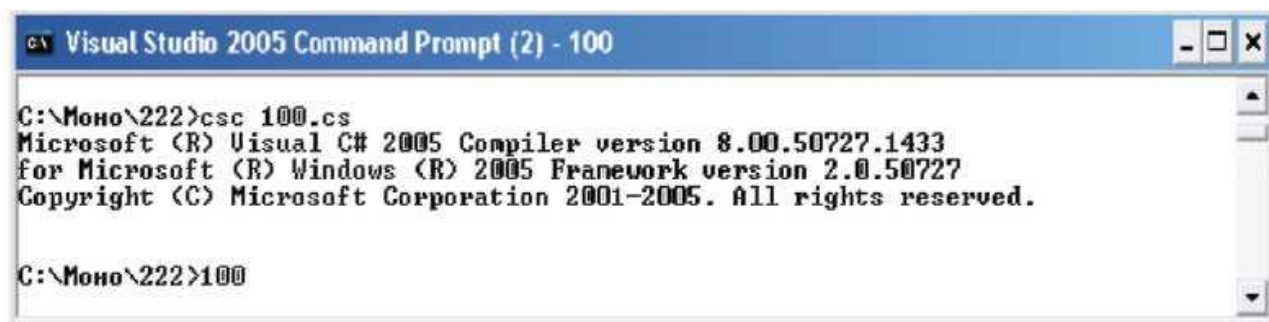
Например, создадим процесс из программы 100.cs, выводящей на экран help программы сетевого клиента ОС - cliconf.chm:

```
using System;
class Class1
{
    [STAThread]
    static void Main(string[] args)
    {
        // Создать новый процесс System.Diagnostics.Process proc = new System.Diagnostics.Process();
        // Приложение которое будем запускать proc.StartInfo.FileName = "cliconf.chm";
    }
}
```

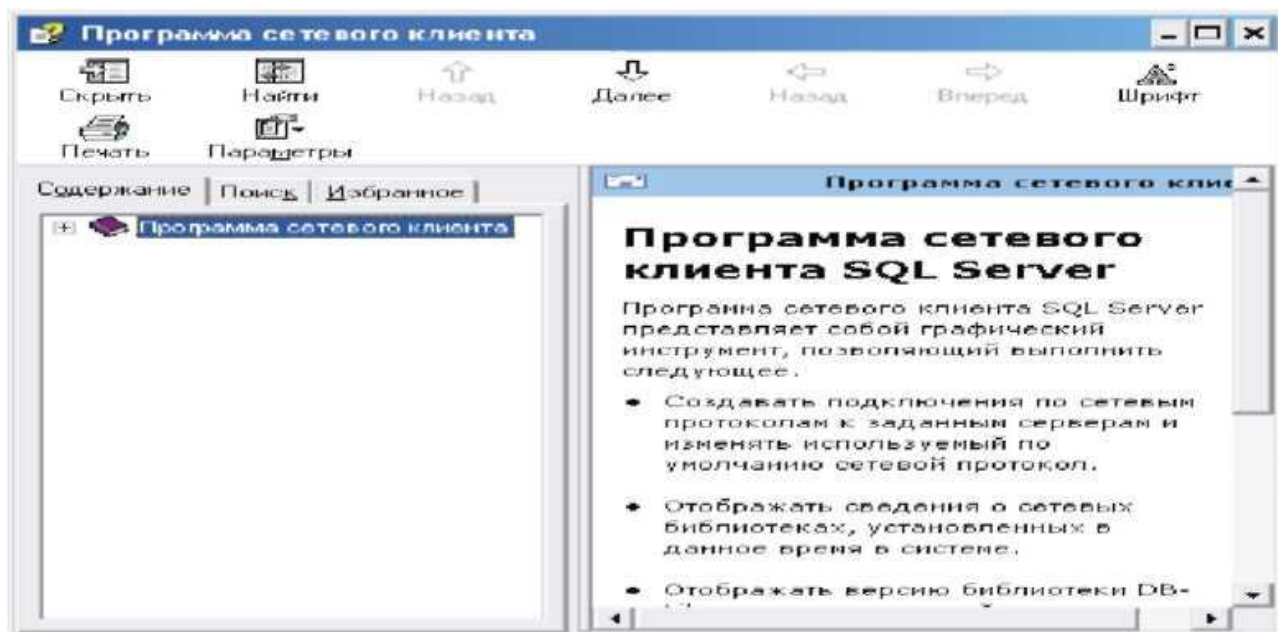
```
proc.Start();  
Console.ReadLine();  
}  
}
```

В этой программе конструктор `System.Diagnostics.Process()` создаёт экземпляр класса `System.Diagnostics.Process` с именем `proc`. Затем определяется требуемый этому процессу ресурс `FileName = " cliconf.chm"`, и запускается метод `Start()`.

Во время компиляции создаётся программа `100.exe`,



запуск которой приводит к появлению соответствующего системного процесса и появлению окна программы сетевого клиента:



Системный процесс - это выполняющееся на компьютере приложение. Из C#-программы доступ к нему можно получить с помощью компонента `Process` (так в MSDN называется экземпляр класса `Process`, создаваемый с помощью конструктора).

Используя компонент `Process`, кроме доступа к системному процессу можно получить список выполняющихся на компьютере процессов или потоков, можно запустить новый процесс, остановить выполняющийся процесс, подключиться к другому процессу, получить информацию о нём: набор потоков, загруженные модули (файлы с расширениями `dll` и `exe`), информацию об объёме занимаемой памяти, временных параметрах выполнения процесса, и т.д.

Для запуска процессов в классе Process может использоваться 6 методов Start:

Start()	Запускает (или начинает заново использовать) ресурс процесса, определенный свойством StartInfo этого компонента Process, и связывает его с компонентом.
Start(ProcessStartInfo)	Запускает ресурс процесса, определенный параметром, содержащим стартовую информацию процесса (например, имя файла процесса для запуска), и связывает ресурс с новым компонентом Process.
Start(String)	Запускает ресурс процесса путем указания имени документа или файла приложения и связывает ресурс с новым компонентом Process.
Start(String, String)	Запускает ресурс процесса путем указания имени приложения и множества аргументов командной строки и связывает ресурс с новым компонентом Process.
Start(String, String, SecureString, String)	Запускает ресурс процесса путем указания имени приложения, имени пользователя, пароля и домена и связывает ресурс с новым компонентом Process.
Start(String, String, String, SecureString, String)	Запускает ресурс процесса путем указания имени приложения, набора аргументов командной строки, имени пользователя, пароля и домена и связывает ресурс с новым компонентом Process.

Среди этих методов только метод Start() не является статическим. Перед его использованием необходимо создать экземпляр класса Process, в свойстве StartInfo надо задать информацию для определения запускаемого ресурса процесса. Причём, в этом свойстве можно задать всё, что можно передать программе в командной строке (обязательным является только значение свойства FileName).

Рассмотрим наиболее важные классы, свойства и методы, используемые в пространстве имён System.Diagnostics.

Свойство FileName

Может определять либо исполнимый файл, либо ресурс для него (т.е. обрабатываемый документ). Что именно должно быть задано этим свойством, уточняется свойством UseShellExecute класса ProcessStartInfo.

Класс ProcessStartInfo

Используется совместно с классом Process. Например, класс Process имеет свойство StartInfo. В MSDN при описании этого свойства говорится, что значение его определяется его типом: System.Diagnostics.ProcessStartInfo. И далее объясняется, что класс ProcessStartInfo представляет данные для запуска процесса. Из этого описания следует, что эти два класса так близки, что в MSDN они почти не различаются.

Класс Process

Имеет свойства, характеризующие заголовок системного процесса, его идентификатор (PID), приоритет, объём занимаемой физической памяти, состав используемых модулей, время начала

и окончания процесса, и другие параметры.

Свойство StartInfo

Позволяет передавать параметры запускаемому процессу. Основным является параметр `FileInfo`. Это может быть исполняемая программа ОС или ресурс.

В составе `StartInfo` в свойстве `Verb` можно указать команду, которую должна выполнить над ресурсом операционная система. Например, при обработке файла, имеющего расширение `.doc`, можно выполнить команду "Print", а при обработке файла с расширением `.exe` можно выполнить команду `runas`.

Состав команд можно определить свойством `Verbs`.

Параметры, указываемые в свойстве `StartInfo`, можно изменить только до момента вызова метода `Start`. После запуска процесса изменение значений `StartInfo` не влияет на запущенный процесс.

Для управления машинными программами представляют интерес такие свойства, как `StandardInput`, `StandardOutput`, `StandardError` (позволяют взаимодействовать со стандартными потоками ввода, вывода и ошибок).

Свойство RedirectStandardError

Позволяет перенаправить поток ошибок, возникающих при выполнении процесса. Можно фильтровать текст, форматировать его, записывать в консоль или предусмотренный для этого файл.

Для изменения направленности потока ошибок свойству `RedirectStandardError` надо присвоить значение `true`, и одновременно необходимо запретить использование средств операционной системы, присвоив свойству `UseShellExecute` значение `false`.

Свойство RedirectStandardInput

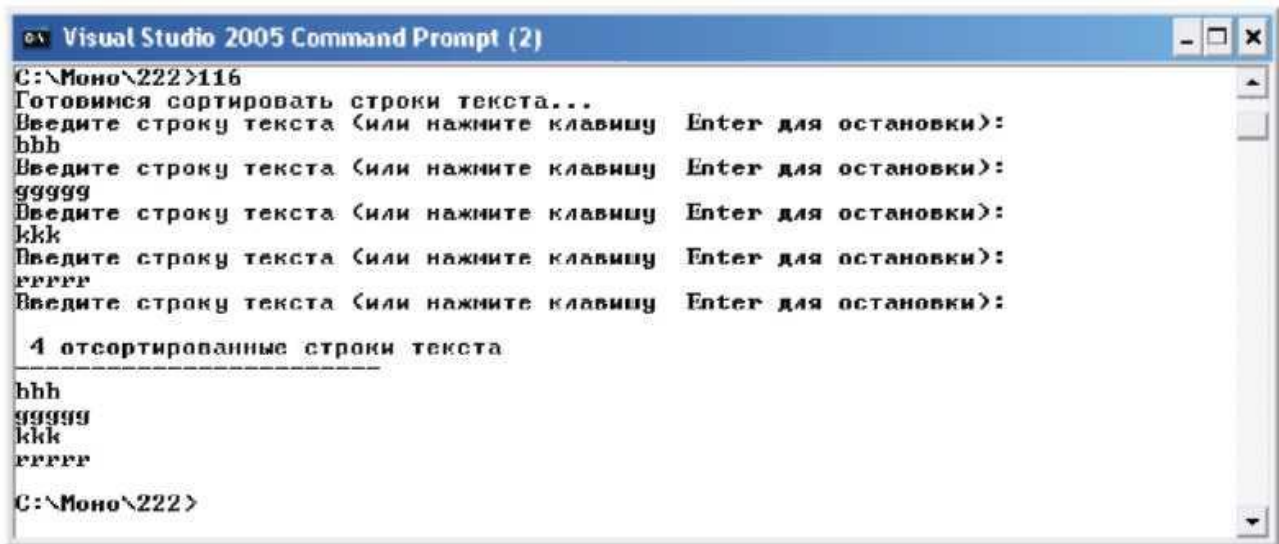
Позволяет перенаправить входной поток.

Стандартным устройством ввода является клавиатура. Его можно заменить на файл - при этом свойству `RedirectStandardInput` присваивается значение `true`.

В следующем примере программой `116.cs` перенаправляется поток процесса `StandardInput` - запускается команда `sort` с перенаправленным вводом. (Команда `sort` является консольным приложением, считывающим и сортирующим введенный текст).

```
using System;
using System.IO;
using System.Diagnostics;
namespace Process_StandardInput_Sample {
class StandardInputTest {
static void Main() {
```

```
Console.WriteLine("Готовимся сортировать строки текста!...");
// Старт процесса Sort.exe с перенаправлением входного потока.
// Команда sort используется для сортировки вводимого текста.
Process myProcess = new Process ( );
myProcess.StartInfo.FileName = "Sort.exe";
myProcess.StartInfo.UseShellExecute = false;
myProcess.StartInfo.RedirectStandardInput = true;
myProcess.Start();
StreamWriter myStreamWriter = myProcess.StandardInput;
// Приглашение ко вводу строки исходного текста для сортировки.
// Записываем каждую строку в поток StandardInput
// с помощью команды sort.
String inputText;
int numLines = 0;
do
{
Console.WriteLine("Введите строку текста (или нажмите клавишу Enter для остановки:");
inputText = Console.ReadLine();
if (inputText.Length > 0)
{
numLines++;
myStreamWriter.WriteLine(inputText);
}
} while (inputText.Length != 0);
// Вывод результата на консоль.
if (numLines > 0)
{
Console.WriteLine (11 {0} отсортированные строки текста 11, numLines);
Console.WriteLine ("-----");
}
else {
Console.WriteLine (11 Нет введённого текста" );
}
// Закрываем входной поток.
// После закрытия потока команда sort
// распечатает отсортированные строки на консоли.
myStreamWriter.Close();
// Ожидаем окончания процесса печати. myProcess.WaitForExit(); myProcess.Close();
}
}
}
```



Если эту программу исполнить без перенаправления, т.е. закомментировать две строки:

```

myProcess.StartInfo.FileName = "Sort.exe";
//myProcess.StartInfo.UseShellExecute = false;
//myProcess.StartInfo.Redirect StandardInput = true;
myProcess.Start();
    
```

то программа 116.exe будет выполняться в окне Visual Studio Command Prompt:



а строки для ввода будут восприниматься из окна запущенной программы sort.exe:



Свойство RedirectStandardOutput

Это свойство позволяет перенаправить выходной поток. Обычно выводимый в выходной поток текст поступает в консоль (выводится на экран). Перенаправление потока эквивалентно командам ">", ">>" или "|" операционной системы. С помощью перенаправления потока можно фильтровать текст, форматировать его, записывать в назначенный файл.

ProcessStandardOutput - это стандартный поток компонента Process.

Свойство RedirectStandardOutput - это перенаправление стандартного потока. Если поток надо перенаправить, то это свойство должно иметь значение true. Если перенаправлять ничего не надо, свойство должно быть равно false.

Совместно с RedirectStandardOutput устанавливается свойство UseShellExecute. При RedirectStandardOutput = true свойство UseShellExecute должно быть false.

Пример:

```
using System;
using System.Diagnostics;
using System.IO;
namespace RunProcess {
class Class1 {
static void Main(string[] args) {
//Структура папки в виде дерева ProcessStartInfo startinfo; Process process = null;
string command, stdoutline;
StreamReader stdoutreader;
// Команда которую будем выполнять command = "tree ";
try {
startinfo = new ProcessStartInfo();
// Запускаем через cmd startinfo.FileName = "cmd.exe";
// Ключ /c - выполнение команды
// Кроме команды можно вставить и необходимые параметры startinfo.Arguments = "/C " +
command + dir;
// Не используем shellexecute startinfo.UseShellExecute = false;
// Перенаправить вывод на обычную консоль
startinfo.RedirectStandardOutput = true;
// Стартуем process = Process.Start(startinfo);
// Получаем вывод stdoutreader = process.StandardOutput;
while((stdoutline=stdoutreader.ReadLine())!= null) {
// Выводим
Console.WriteLine(stdoutline);
}
stdoutreader.Close(); stdoutreader = null;
}
catch {
throw;
}
finally {
if (process != null) {
// Закрываем process.Close();
}
// Освобождаем process = null; startinfo = null;
}
Console.ReadLine();
}
}
[STAThread]
static void Main(string[] args) {
```

```
string dir = args[0]; structura(dir); }  
}  
}
```



Свойство StandardOutputEncoding

Возвращает или задает предпочтительную кодировку для стандартного вывода.

Если значение этого свойства равно null, процесс использует для стандартного вывода соответствующую кодировку по умолчанию.

Свойство ProcessStartInfo.UseShellExecut

Возвращает или задает значение, позволяющее определить, нужно ли использовать оболочку операционной системы для запуска процесса.

Для использования оболочки системы при запуске процесса, свойство должно иметь значение true. В противном случае процесс создается непосредственно из исполняемого файла, а возможности ОС не используются.

При использовании оболочки ОС запускать процесс можно по имени обрабатываемого документа. По этому имени оболочка ОС определяет тип документа и находит в реестре, какая программа является обработчиком документов данного типа, после чего запускает эту программу и передает ей в качестве параметра обрабатываемый файл.

Кроме того, при использовании оболочки ОС файл может в некоторых случаях задаваться без указания пути к нему. Если путь не указан, файл сначала ищется в текущем каталоге, а при отрицательном исходе - в каталогах, перечисленных в системной переменной path.

К особенностям использования оболочки ОС относится и способность ОС определять приоритет запуска файлов, имеющих

одинаковые имена, но разные расширения. Например, в текущем каталоге хранится три файла: "a.cmd", "a.com" и "a.exe". Если процесс запускается без указания расширения, только по имени файла, при UseShellExecute = true операционная система сначала выполнит файл с расширением ".com", затем - ".exe", и только затем - ".cmd".

При UseShellExecute = false файлы будут выполняться в том порядке, в котором они были записаны в каталог.

Задав для этого свойства значение false, можно перенаправлять потоки ввода, вывода и ошибок.

Свойство ProcessStartInfo.Verb

Возвращает или задает команду, которую необходимо использовать при открытии приложения или документа, указанного в свойстве FileName.

Каждое расширение файла имеет свой набор команд (этот набор можно получить с помощью свойства Verbs). Примерами команд могут служить "Print", "Edit", "Open", "OpenReadOnly" и другие.

При использовании свойства Verb в значении свойства FileName обязательно нужно указывать расширение файла, так как именно по расширению определяется допустимый состав команд.

Свойство ProcessStartInfo.Verbs

Возвращает набор команд, которые связаны с типом файла, заданным в свойстве FileName.

Пример:

В следующем примере кода выводятся команды, заданные для выбранного типа файла, определяемого его расширением.

Пример:

```
//Программа 102 .exe, определяющая состав допустимых команд
//для заданного типа файла.
using System;
using System.IO;
using System.Diagnostics;
using System.Text;
using System.Windows.Forms;
class ProcessInformation {
    [STAThread]
    static void Main(string[] args) {
        // Создать новый процесс
        System.Diagnostics.Process proc = new System.Diagnostics.Process();
        // Приложение которое будем запускать proc.StartInfo.FileName = "cmd.exe";
        proc.Start();
        int i = 0;
        foreach (String verb in proc.StartInfo.Verbs) {
            // Display the possible verbs. Console.WriteLine(" {0}. {1}", i.ToString(), verb); i++;
        }
        Console.ReadLine();
    }
}
```

Для файла .jpg допустимыми будут команды:

```

Visual Studio 2005 Command Prompt (2)
C:\Моно\222>102
 0. ACDEdit
 1. ACDPrint
 2. ACDView
 3. open
 4. printto
C:\Моно\222>
    
```

При запуске программы 102.exe для файла cmd.exe получаем:

```

Visual Studio 2005 Command Prompt (2)
C:\Моно\222>102
 0. open
 1. runas
C:\Моно\222>_
    
```

Запрашиваем help команды runas:

```

Visual Studio 2005 Command Prompt (2)
C:\Моно\222>runas /?
Использование команды RUNAS:

RUNAS [ [/noprofile : /profile] [/env] [/netonly] ]
        /user:<имя пользователя> <программа>

RUNAS [ [/noprofile : /profile] [/env] [/netonly] ]
        /smartcard [/user:<имя пользователя>] <программа>

/noprofile      Не загружать профиль пользователя. Это приводит к более
                быстрой загрузке приложения, но может стать причиной
                неправильной работы некоторых приложений.
/profile        Загружать профиль пользователя.
                Этот параметр установлен по умолчанию.
/env           Использовать текущие параметры среды.
/netonly       Учетные данные предназначены только для удаленного
                доступа.
/savescred     Использовать учетные данные, сохраненные пользователем.
                Этот параметр не доступен в Windows XP Home Edition
                и будет проигнорирован.
/smartcard     Для указания учетных данных используется
                смарт-карта.
/user <имя пользователя> должно быть в виде USER@DOMAIN или DOMAIN\USER
<программа>   Командная строка для EXE. См. примеры ниже.

Примеры:
> runas /profile /user:mymachine\administrator cmd
> runas /profile /env /user:mydomain\admin "mmc %windir%\system32\dsa.nsc"
> runas /env /user:user@domain.microsoft.com "notepad \"Мой файл.txt\"

Примечание: вводите пароль пользователя только тогда, когда он запрашивается.
Примечание: формат записи USER@DOMAIN несовместим с параметром /netonly.
Примечание: параметр /profile несовместим с параметром /netonly.
C:\Моно\222>
    
```

Свойство ProcessStartInfo.WorkingDirectory

Возвращает или задает начальный каталог для запускаемого процесса.

Если свойство не задано, по умолчанию рабочим каталогом является %SYSTEMROOT%\system32.

Если значением свойства является пустая строка, исполняемый файл находится в текущем каталоге.

СОЗДАНИЕ ПРОЦЕССОВ И УПРАВЛЕНИЕ ИМИ

Кириченко Александр Аполлонович

Запуск командного процессора

Для запуска 32-разрядного командного процессора операционной системы создаём процесс с идентификатором `tu`, в свойстве `FileName` указываем полный адрес командного процессора и запускаем процесс:

```
using System;
using System.Diagnostics;
class MyProcess
{
public static void Main()
{
Process my = new Process();
my.StartInfo.FileName = "C:\\WINDOWS.0\\system32\\cmd.exe"
my.Start();
Console.Read(); }
}
```

В результате появляется окно командного процессора:



Ресурсы процесса и передача параметров запускаемому процессу

В операционной системе Windows запустить процесс можно двумя способами:

1. Запустить обрабатывающую программу и с помощью параметров передать этой программе имя обрабатываемого документа;
2. Дважды щёлкнуть по файлу, содержащему обрабатываемый документ. Операционная система по расширению файла определит, какая программа обрабатывает документ такого типа, вызовет на исполнение эту программу и передаст ей обрабатываемый документ.

Вызов программы: указанием ресурса.

C#-программа `processstartinfo2.exe` используется для запуска файла, содержащего исходный код программы. Запуск процесса производится указанием имени запускаемого ресурса. Поскольку

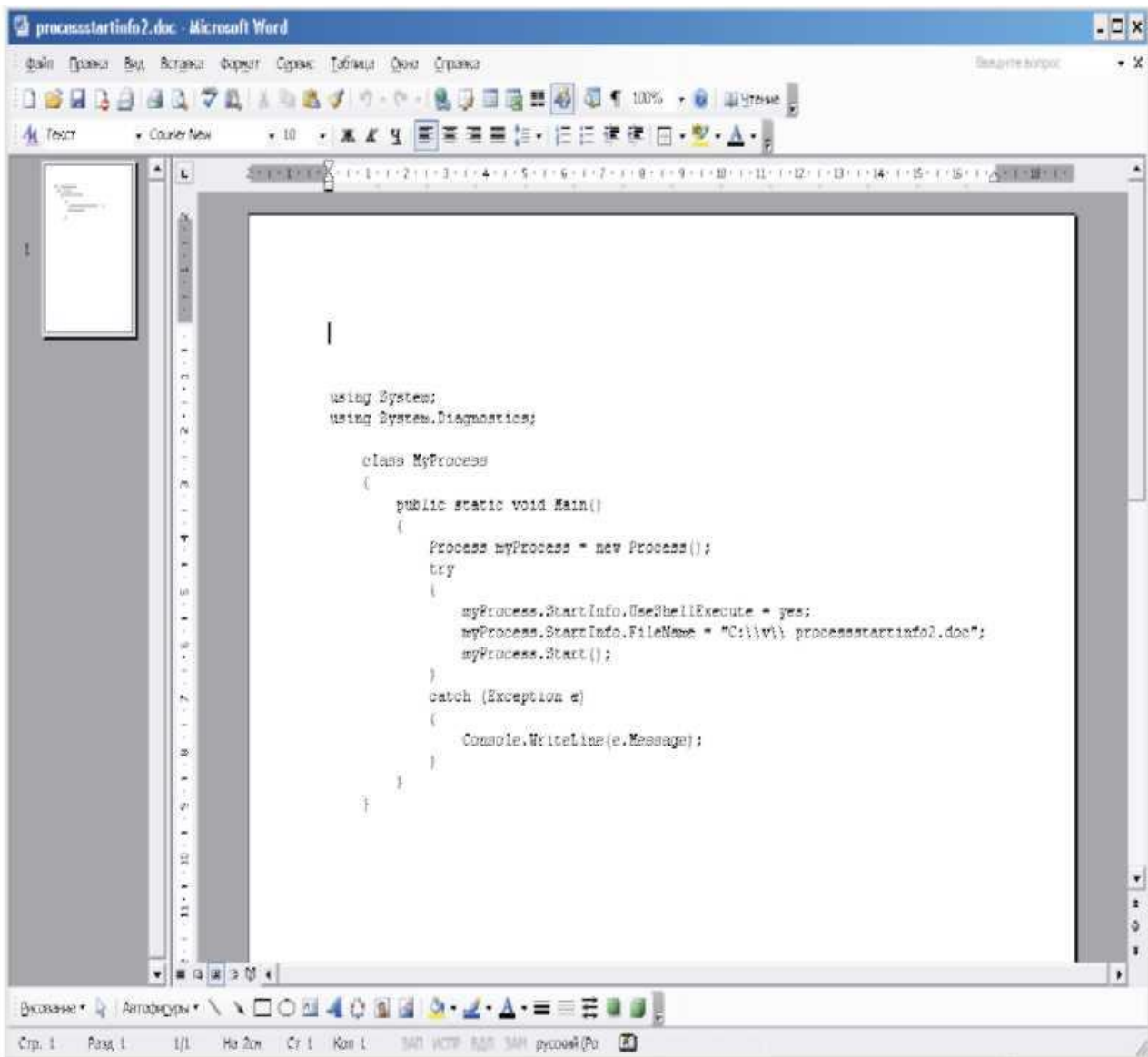
имя программы для обработки данного ресурса не указывается, необходимо разрешить операционной системе использовать свои возможности. Для этого свойство StartInfo.UseShellExecute получает значение true:

```
using System;
using System.Diagnostics;
class MyProcess
{
public static void Main()
{
Process myProcess = new Process();
try
{
myProcess.StartInfo.UseShellExecute = true; myProcess.StartInfo.FileName =
"C:\\v\\processstartinfo2.doc";
myProcess.Start();
}
catch (Exception e) {
Console.WriteLine(e.Message);
}
}
}
```

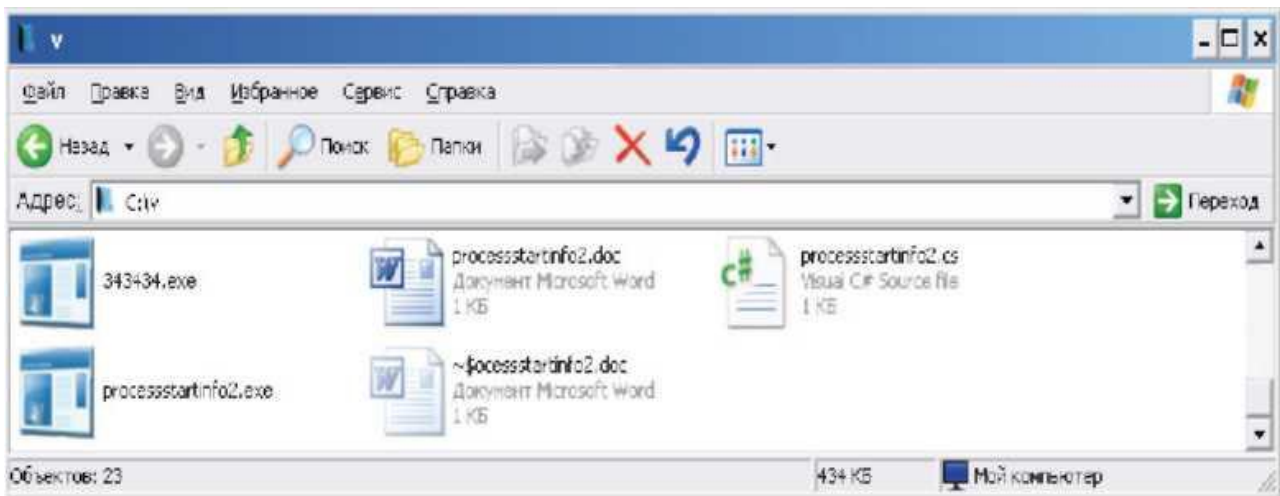
Все необходимые файлы находятся в папке v. Протокол запуска программы:

```
C:\v>processstartinfo2
C: \v>
```

На экране появляется окно Word, содержащее исходный код программы processstartinfo2.es:



Фрагмент окна папки v:



В данном случае C#-программа была запущена двойным щелчком по ресурсу (другими словами - по документу, который должен быть открыт).

Перехват результатов работы процесса

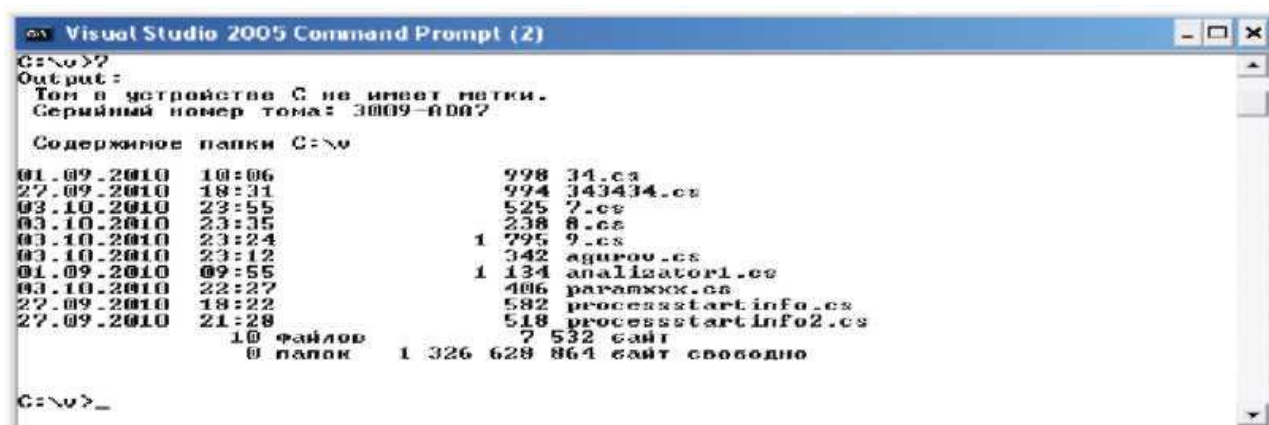
Процесс создаётся для того, чтобы выполнить какую-то часть работы, а затем использовать полученный результат. С этой целью выполним программу 7.cs:

```
//Программа 7.cs
using System;
using System.Diagnostics;
class MainClass {
public static void Main() {
Process p = new Process(); p.StartInfo.FileName = "cmd.exe"; p.StartInfo.Arguments = "/c dir *.cs";
p.StartInfo.UseShellExecute = false; p.StartInfo.RedirectStandardOutput = true;
p.Start();
string output = p.StandardOutput.ReadToEnd();
Console.WriteLine("Output:"); Console.WriteLine(output); }
}
```

В этой программе с помощью 32-разрядного процессора cmd.exe запрашивается, какие исходные тексты программ, написанных на языке с# содержатся в текущем директории. Имеется в виду директорий v: в корне диска C:.

В свойстве StartInfo.Arguments содержится комбинация символов /c, обозначающая необходимость выполнения команды операционной системы - далее указывается, что такой командой должна быть команда dir. Результат выполнения этой команды разрешено перенаправить в стандартный выходной поток.

Последние три команды посвящены работе с этим потоком. Сначала стандартный выходной поток направляется в переменную output. Затем выводится на печать заголовок, предупреждающий, что будет выводиться содержимое стандартного выходного потока, т. е. результат работы команды dir операционной системы. И затем распечатывается содержание переменной output, т. е. список имеющихся в каталоге у исходных текстов сопрограмм.



```
Visual Studio 2005 Command Prompt (2)
C:\>?
Output:
Топ в устройстве C не имеет метки.
Серийный номер тома: 3809-0002

Содержимое папки C:\v

01.09.2010 10:06          998 31.cs
27.09.2010 18:31          994 343434.cs
03.10.2010 23:55          525 7.cs
03.10.2010 23:15          238 8.cs
03.10.2010 23:24           1 795 9.cs
03.10.2010 23:12           1 342 agurov.cs
01.09.2010 09:55           1 134 analisatori.cs
03.10.2010 22:27          486 paramxxx.cs
27.09.2010 18:22          582 processstartinfo.cs
27.09.2010 21:28          518 processstartinfo2.cs
           10 файлов          7 532 байт
           0 папок          1 326 628 864 байт свободно

C:\>_
```

ЗАПУСК КОМАНД ОС ИЗ C#-ПРОГРАММЫ

Кириченко Александр Аполлонович

Установка атрибутов файла

Получить информацию об атрибутах файла с помощью команд ОС можно, указав в свойстве StartInfo.Arguments команду attrib и имя файла:

```
using System; using System.Diagnostics;
class MainClass {
public static void Main() {
Process p = new Process();
p.StartInfo.FileName = "cmd.exe";
p.StartInfo.Arguments = "/c attrib 104.exe";
p.StartInfo.UseShellExecute = false;
p.StartInfo.RedirectStandardOutput = true;
p.Start();
string output = p.StandardOutput.ReadToEnd(); Console.WriteLine("Output:");
Console.WriteLine(output); }
}
```



```
Visual Studio 2005 Command Prompt (2)
C:\Моно\222>105
Output:
A C:\Моно\222\104.exe
C:\Моно\222>_
```

Полученная информация говорит о том, что файл 104.exe имеет атрибут A (архивный).

Если изменить свойство StartInfo.Arguments на "/c attrib +S 104.exe", присвоим файлу ещё и атрибут S - 'системный':



```
Visual Studio 2005 Command Prompt (2)
C:\Моно\222>105
Output:
C:\Моно\222>attrib 104.exe
A S C:\Моно\222\104.exe
C:\Моно\222>_
```

При этом сам процесс присвоения никак не отражается в листинге: программа 105.exe в

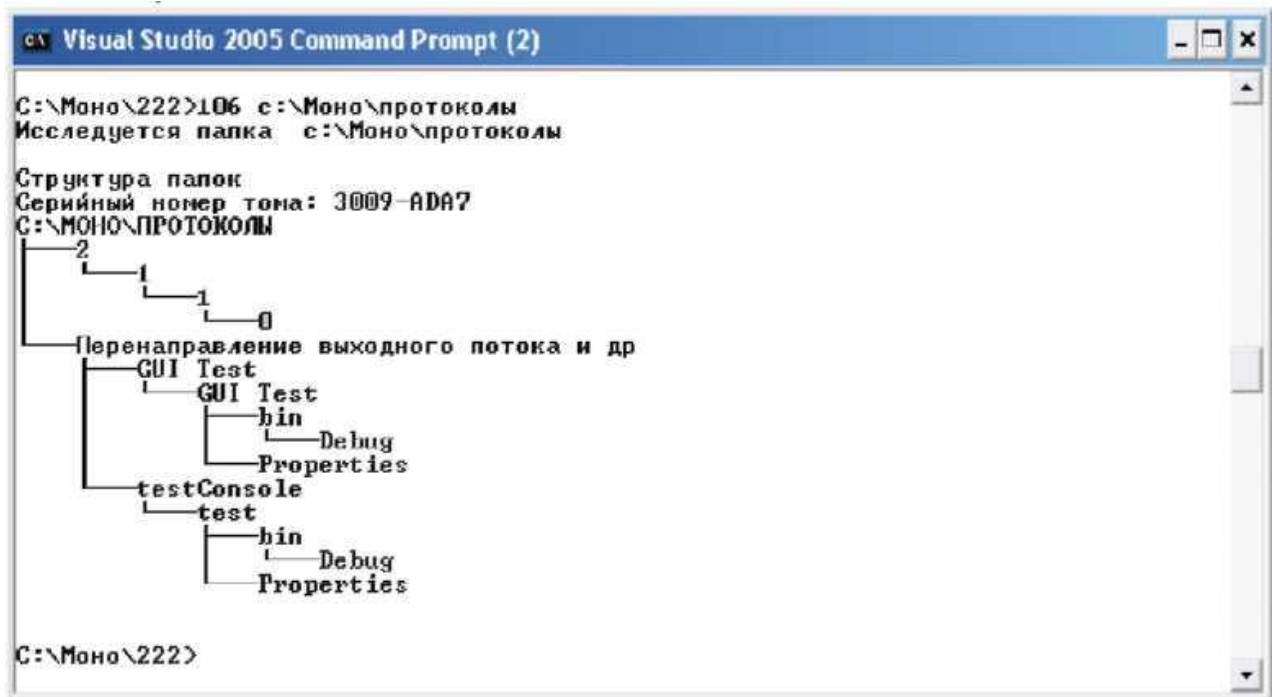
выходной поток ничего не направляет.

Получение структуры папки по команде tree

Из C#-программы можно получить структуру любой папки в виде дерева с помощью команды операционной системы tree. Воспользуемся для этого программой 106.cs.

```
//Программа 106.cs: Структура папки в виде дерева.
//Имя папки указать в командной строке при запуске программы.
using System;
using System.Diagnostics;
using System.IO;
class Class1 {
static void Main(string[] args) {
string dir = args[0]; Console.WriteLine("Исследуется папка " + dir + "\n");
ProcessStartInfo startinfo;
Process process = null;
string command, stdoutline;
StreamReader stdoutreader;
// Команда которую будет выполнять command = "tree ";
try
{
startinfo = new ProcessStartInfo();
// Запускаем через cmd startinfo.FileName = "cmd.exe";
// Ключ /c - выполнение команды
// Кроме команды можно вставить и необходимые параметры startinfo.Arguments = "/C " +
command + dir;
// Не используем shellexecute startinfo.UseShellExecute = false;
// Перенаправить вывод на обычную консоль
startinfo.RedirectStandardOutput = true;
// Стартуем process = Process.Start(startinfo);
// Получаем вывод stdoutreader = process.StandardOutput;
while((stdoutline=stdoutreader.ReadLine())!= null)
{
// Выводим
Console.WriteLine(stdoutline);
}
stdoutreader.Close(); stdoutreader = null;
}
catch
{
throw;
}
finally
{
if (process != null)
{
// Закрываем process.Close();
}
}
}
}
```

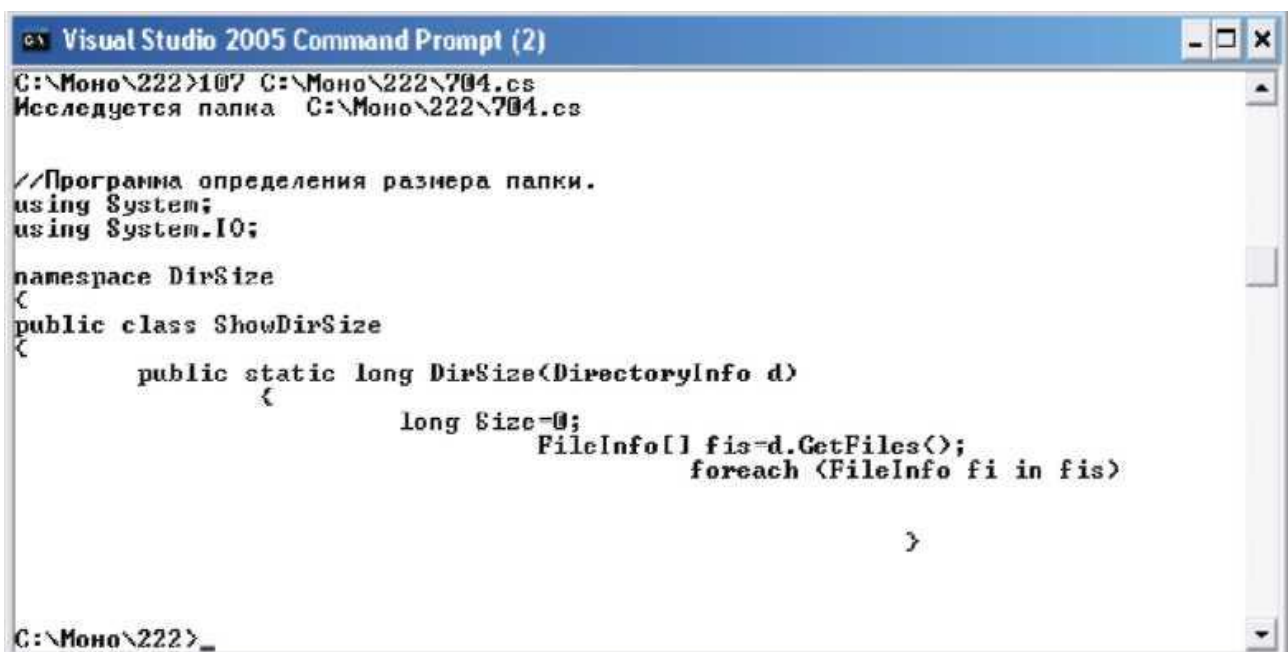
```
// Освобождаем process = null; startinfo = null;  
}  
Console.ReadLine();  
}  
}
```



```
Visual Studio 2005 Command Prompt (2)  
C:\Моно\222>106 c:\Моно\протоколы  
Исследуется папка c:\Моно\протоколы  
  
Структура папок  
Серийный номер тома: 3009-ADA7  
C:\МОНО\ПРОТОКОЛЫ  
├── 2  
│   ├── 1  
│   │   └── 1  
│   │       └── 0  
│   └── Перенаправление выходного потока и др  
├── GUI Test  
│   ├── GUI Test  
│   │   ├── bin  
│   │   │   └── Debug  
│   │   └── Properties  
└── testConsole  
    └── test  
        ├── bin  
        │   └── Debug  
        └── Properties  
  
C:\Моно\222>
```

Вывод на экран текстового файла командой type

Если в программе 106.cs вместо команды tree вставить команду type, получим вывод на экран текста указанного в командной строке файла ():



```
Visual Studio 2005 Command Prompt (2)  
C:\Моно\222>107 C:\Моно\222\704.cs  
Исследуется папка C:\Моно\222\704.cs  
  
//Программа определения размера папки.  
using System;  
using System.IO;  
  
namespace DirSize  
{  
    public class ShowDirSize  
    {  
        public static long DirSize(DirectoryInfo d)  
        {  
            long Size=0;  
            FileInfo[] fis=d.GetFiles();  
            foreach (FileInfo fi in fis)  
                Size+=fi.Length;  
            return Size;  
        }  
    }  
}
```

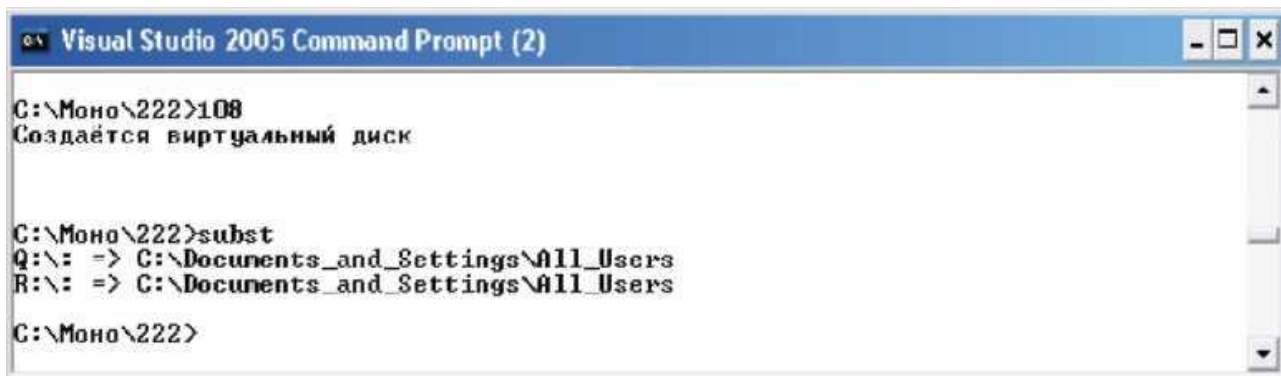
Создание виртуального диска

Используем программу 108.cs и команду subst

Надо только иметь в виду, что команда type выводит текст в коде ASCII. Для вывода текста в другой кодировке необходимо программу вывода перестроить операционной системы для создания виртуального диска Q:.

```
//Программа 108.cs. Создание виртуального диска
using System;
using System.Diagnostics;
using System.IO;
class Class1
{
static void Main(string[] args) {
Console.WriteLine("Создаётся виртуальный диск " + "\n");
ProcessStartInfo startinfo; Process process = null;
string command, stdoutline;
StreamReader stdoutreader;
// Команда которую будет выполнять операционная система
command = "subst q: ";
try {
startinfo = new ProcessStartInfo();
// Запускаем через cmd startinfo.FileName = "cmd.exe";
// Ключ /c - выполнение команды
//Кроме команды можно вставить и необходимые параметры
// Задаём папку, в которой будет создан виртуальный диск
string dir = @"C:\Documents_and_Settings\All_Users";
startinfo.Arguments = "/C " + command + dir;
// Не используем shellexecute startinfo.UseShellExecute = false;
// Перенаправить вывод на обычную консоль
startinfo.RedirectStandardOutput = true;
// Стартуем process = Process.Start(startinfo);
// Получаем вывод stdoutreader = process.StandardOutput;
while((stdoutline=stdoutreader.ReadLine())!= null) {
// Выводим
Console.WriteLine(stdoutline);
}
stdoutreader.Close();
stdoutreader = null;
}
catch
{
throw;
}
finally
{
if (process != null)
{
// Закрываем process.Close();
}
```

```
}  
// Освобождаем process = null; startinfo = null;  
}  
Console.ReadLine();  
}  
}
```



```
Visual Studio 2005 Command Prompt (2)  
C:\Моно\222>108  
Создается виртуальный диск  
  
C:\Моно\222>subst  
Q:\: => C:\Documents_and_Settings\All_Users  
R:\: => C:\Documents_and_Settings\All_Users  
  
C:\Моно\222>
```

Проверка показывает, что в заданной папке создан виртуальный диск Q:.

Управление работой архиватора

Кроме программы WinRAR, имеющей оконный интерфейс, в комплект поставки архиватора входит файл Rar.exe. Это также 32-разрядная версия RAR для Windows, но она поддерживает только интерфейс командной строки и работает в текстовом (консольном) режиме. Консольную версию RAR удобно использовать для вызова из пакетных файлов (BAT и CMD), для запуска из приглашения DOS и т.п. Она поддерживает больше команд и ключей в командной строке, чем WinRAR.

Общий синтаксис командной строки для архивации файлов таков:

```
Rar.exe A [-ключи ] <Архив> [ Файлы ] [ @Файлы-списки ]
```

Например, если необходимо добавить файл copycat1.cs в архив LETTER.RAR, используется команда:

```
rar.exe a letter.rar copycat1.cs
```

Для сохранения в архиве файла используем программу 109.cs:

```
//109.cs: Управление архиватором из C#-программы  
using System;  
using System.Diagnostics;  
using System.IO;  
class Class1  
{  
static void Main(string[] args) {  
ProcessStartInfo startinfo;
```

```
Process process = null;
string command;
// Команда которую будет выполнять command = " a ";
try {
string arhiv = "letter.rar"; string file = "copycat1.cs";
startinfo = new ProcessStartInfo();
// Запускаем архиватор startinfo.FileName = "rar.exe";
// Ключ /c - выполнение команды
// Кроме команды можно вставить и необходимые параметры
startinfo.Arguments = "/C " + command + " " + arhiv + " " + file;
// Не используем shellexecute startinfo.UseShellExecute = false;
// Стартуем process = Process.Start(startinfo); }
catch {
throw;
}
finally {
if (process != null) {
// Закрываем process.Close();
}
// Освобождаем process = null; startinfo = null;
}
Console.ReadLine();
}
}
```



```
Visual Studio 2005 Command Prompt (2)
C:\Мано\222>109
RAR 3.51 Copyright (c) 1993-2005 Alexander Roshal 7 Oct 2005
Shareware version Type RAR -? for help
Evaluation copy. Please register.
Creating archive letter.rar
Adding copycat1.cs OK
Done
C:\Мано\222>_
```

При старте программы 109.exe запускается программа rar.exe и сообщает через консоль, что создаётся архив letter.rar и в него добавляется файл copycat1.cs. Операция выполнена успешно, о чём свидетельствуют OK Done.

Создание командного файла из C#-программы.

Создание текстового файла (bat или cmd)

Для того чтобы создать командный файл необходимо создать поток для записи в файл.

```
StreamWriter s = new StreamWriter("1.cmd", false);
```

где "1.cmd" - это имя создаваемого файла;

false - определяет, требуется ли добавить в файл данные. Если файл существует и значение этого параметра равно false, файл перезаписывается; если не существует, создается новый файл.

После создания потока необходимо записать в файл соответствующие команды. Запись в файл производится командами класса System.Console, такими, как Write() или WriteLine(). Например, создать директорию с именем "1" можно по команде операционной системы md 1. Перед командой надо указать имя созданного потока:

```
s.WriteLine("md 1");
```

Вывод на экран созданного файла

Выведем на экран содержимое нашего командного файла. Для этого создаем поток для чтения из файла и построчно выводим на экран его содержимое.

```
StreamReader r = new StreamReader("1.cmd");  
while (!r.EndOfStream)  
{  
    string str = r.ReadLine(); Console.WriteLine(str);  
}  
r.Close();
```

Здесь EndOfStream свойство класса StreamReader, определяющее, находится ли позиция текущего потока в конце потока. Пример.

В папке C:\Моно\222> размещаем файл 110.cs, в котором предусмотрено создание командного файла операционной системы, содержащего всего одну команду: "md 1", а затем - чтение этого файла с выводом его содержимого на экран.

```
//Файл 110.cs:  
using System;  
using System.IO;  
public class Show  
{  
    public static void Main() {  
        StreamWriter s = new StreamWriter("1.cmd",  
            false);  
        s.WriteLine("md 1"); s.Close();  
        StreamReader r = new StreamReader("1.cmd");  
        while (!r.EndOfStream)  
        {  
            string str = r.ReadLine(); Console.WriteLine(str);  
        }  
        r.Close();  
        Console.Read(); }  
}
```

После компиляции получаем файл 110.exe, при исполнении которого на экран выводится созданная команда:



```
C:\Моно\222>110
nd 1

C:\Моно\222>_
```

В папке с программой 110.cs появился файл 1.cmd:



Проблема длинных имён и возможные пути её решения

В операционной системе Windows разрешено использование «длинных имён» файлов и папок (каталогов или директориев). По данным MSDN полная длина имени файла не должна превышать 260 символов, а длина пути до файла или папки не должна быть выше 248 символов.

Однако во время работы каталоги формируются не сразу, а постепенно. Из них выстраивается иерархическая структура, в которой общая длина имени папок и файлов может значительно превысить установленные пределы, так как длины имён файлов и папок стандартными средствами операционной системы не контролируются.

Если длина адреса (пути к файлу) превышает допустимые пределы, представим полный путь к файлу (P) состоящим из двух частей: p1 (допустимая длина пути) и p2 (запредельный путь).

Операционная система файл, находящийся по адресу P из корня диска не видит. Поэтому ни одна команда операционной системы с этим файлом работать не сможет, если его попытаться достать из активного каталога, находящегося близко к корню диска.

Добраться до этого файла можно, если переместить активный каталог ОС на границу

допустимого и запредельного пути. В этом случае путь к файлу нужно задавать относительно активного каталога, так как полный адрес к файлу распадётся на две части -р1 и р2:

```
C:\р1>сору р2
```

Допустимый путь р1 указывает операционной системе, откуда надо начинать выполнение команды, а тот адрес, который раньше был запредельным, становится допустимым.

Эта особенность работы с длинными именами стандартными средствами операционных систем Windows не учитывается. Поэтому часто можно встретиться с отказом работы с некоторыми папками Windows программами - архиваторами, программами для переноса информации на компакт-диски и DVD-диски, и др.

При копировании или перемещении такой папки надо либо использовать нестандартную программу копирования (перемещения), которая будет делить полный путь на два: допустимый - и запредельный, и переносить активный каталог на границу этих двух миров, либо надо корректировать структуру такого неправильного каталога.

Коррекция структуры выгоднее, так как позволит с исправленным каталогом работать стандартными средствами.

Коррекцию можно произвести по-разному. Рассмотрим два возможных метода: метод 1 - «сжатие имён»; метод 2 - архивирование «запредельной части» (и таким образом - сжатие полного имени).

Для первого метода можно применить один из известных алгоритмов сжатия (например - Лемпеля-Зива, DES, и др.). Но для работы полученные новые имена нужно как-то связывать со старыми...

Для второго метода структура информации практически не изменяется. Нужно только применить какой-нибудь архиватор.

При реализации 2 метода задача будет заключаться в следующем:

1. Начать копирование (перемещение) заданной папки стандартным образом.
2. При возникновении исключения из-за длины пути, выделить из полного пути допустимый путь и в конец его перенести активный каталог (в операционной системе это делается с помощью команды cd).
3. Переместить каталог, вызвавший исключение, в корень диска. Операционная система это позволит сделать, так как запредельный ранее каталог теперь стал доступным.
4. Архиватором сжать перемещённый каталог.
5. Полученный архив переместить на место, которое занимал запредельный каталог.
6. Продолжить операцию копирования (перемещения).

Результатом этого будет откорректированный каталог, в котором запредельно длинных имён не будет. Такой каталог можно скопировать или переместить стандартными средствами ОС, а если нужно сохранить прежнюю структуру каталога, архив можно сделать саморазворачивающимся (SFX) и развернуть его после копирования на новое место.

Для ускорения работы архиватор можно использовать стандартный, например, RAR.

В данном случае решение задачи достигается за счёт комплексирования программных средств: C#-программы копирования (перемещения) каталогов, операторов ОС и посторонней

программы архивации.

Проверка наличия в папке имён, длиной более 200 символов

При запуске программы в командной строке после имени запускаемой программы указывается имя исследуемой папки. Например, имя программы 114.exe, исследуется папка C:\Моно\Результат. Текст программы 114.cs:

```
//Программа 114.cs. Фиксация папок с длиной имени более 200 символов:
using System;
using System.IO;
class Class1 {
//Точка входа в программу [STAThread]
static void Main(string[] args) {
string dir = args[0]; analiz(dir);
}
static void analiz(string dir) //Анализ длины имён
{
Console.WriteLine("Исследуется папка " + dir + "\n");
string [] subdirectory =
Directory.GetDirectories(dir,"*",System.IO.SearchOption.AllDirectories);
Console.WriteLine("Количество подкаталогов в папке " + subdirectory.Length);
int k=0;
for (int i = 0; i < subdirectory.Length; i++)
{
if(subdirectory[i].Length > 200)
{
Console.WriteLine(i + " " + subdirectory[i] + " " + subdirectory[i].Length); k+= 1;
}
}
if(k==0)
{
Console.WriteLine("Папок с длиной имени более 200 символов нет ");
}
}
}
```

При отсутствии длинных имён:

```

Visual Studio 2005 Command Prompt (2)
C:\Моно\222>114 "C:\Documents and Settings"
Исследуется папка C:\Documents and Settings

Количество подкаталогов в папке 3336
Папок с длиной имени более 200 символов нет

C:\Моно\222>

```

А при наличии длинных имён выводятся порядковый номер каталога, путь к нему и длина имени:

```

Visual Studio 2005 Command Prompt (2)
C:\Моно\222>114 "C:\Documents and Settings"
Исследуется папка C:\Documents and Settings

Количество подкаталогов в папке 3466
92 C:\Documents and Settings\Пользователь\Мои документы\Visual Studio 2005\Proj
ects\All Users.WINDOWS.0\Документы\SAА\Эталон\Эксперимент\Клавиатурный почерк че
ловека\Клавиатурный почерк человека\Properties 202
93 C:\Documents and Settings\Пользователь\Мои документы\Visual Studio 2005\Proj
ects\All Users.WINDOWS.0\Документы\SAА\Эталон\Эксперимент\Клавиатурный почерк че
ловека\Клавиатурный почерк человека\obj\Debug 201
94 C:\Documents and Settings\Пользователь\Мои документы\Visual Studio 2005\Proj
ects\All Users.WINDOWS.0\Документы\SAА\Эталон\Эксперимент\Клавиатурный почерк че
ловека\Клавиатурный почерк человека\obj\Debug\TempPE 208
99 C:\Documents and Settings\Пользователь\Мои документы\Visual Studio 2005\Proj
ects\All Users.WINDOWS.0\Документы\SAА\Эталон\++2010\3\2009н\Архитектура интелле
ктуальных программных систем\Семинар\Распознавание 206
100 C:\Documents and Settings\Пользователь\Мои документы\Visual Studio 2005\Pro
jects\All Users.WINDOWS.0\Документы\SAА\Эталон\++2010\3\2009н\Архитектура интелл
ектуальных программных систем\Семинар\Распознавание\Клавиатурный почерк @ 228
186 C:\Documents and Settings\Пользователь\Мои документы\Visual Studio 2005\Pro
jects\All Users.WINDOWS.0\Application Data\SolidDocuments\Installer\Solid Conver
ter PDF\Администратор\SolidSFX_Data\components 201

C:\Моно\222>

```

Программа коррекции неправильных каталогов (со сверхдлинными путями)

Коррекция каталогов со сверхдлинными именами предусматривает:

1. Создание в корне какого-либо диска рабочего каталога с именем "v".
2. Размещение в этом каталоге необходимых для работы файлов: основного файла -архиватора gar.exe, вспомогательного файла архиватора - default.sfx, командного файла для запуска архиватора - comand.cmd.
3. При моделировании реконструкции каталога требуется дополнительное мероприятие: создание каталога требуемой конструкции. Такой каталог может быть создан по адресу: c:\Эксперимент\3. Этот каталог должен содержать иерархическую структуру подкаталогов с

требуемой длиной и конструкцией имён.

4. Программа `remont.cs` выполняет следующие действия:

1. получает структуру каталогов исследуемой папки;
2. перемещает каталог `subdir` (`subdir` - это переменная, содержащая текущее значение имени), длина имени которого превышает заданную величину, в рабочий каталог `v` и изменяет его имя на `1`;
3. запускает файл `command.cmd`, преобразующий каталог в `sfx`-архив (файл);
4. перемещает созданный `sfx`-архив на место каталога `subdir` и изменяет его имя на `subdir.exe`.

Если каталог `subdir` имел в свою очередь иерархическую структуру, она обрывается, и таким образом сокращается общая длина имени.

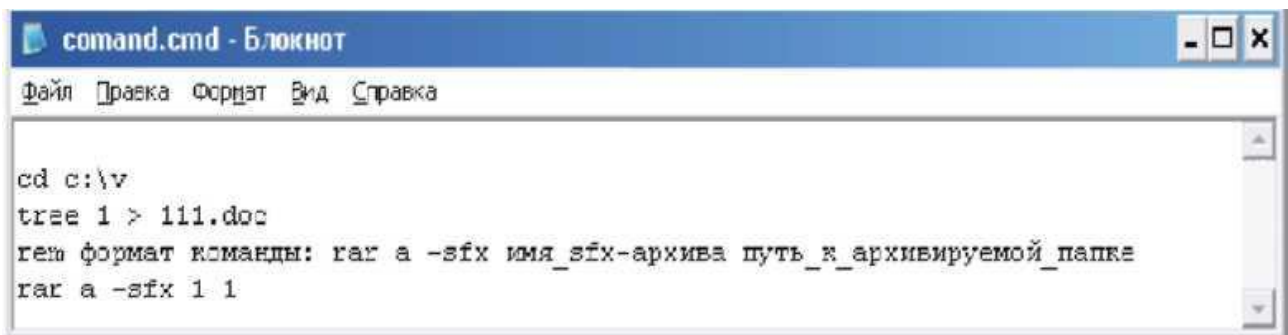
```
//Текст программы Remont2.cs
using System;
using System.IO;
using System.Diagnostics;
namespace RemontDir
{
class RemDir {
static void Main(string[] args)
{
string rab = @"c:\v\1";
string dir = @"C:\y\Эксперимент\3";
string [] subdirectory = Directory.GetDirectories(dir,"*",System.IO.SearchOption.AllDirectories);
foreach (string subdir in subdirectory)
{
if (subdir.Length > 200) {
Console.WriteLine("\n" + subdir + " " + subdir.Length);
Directory.Move(subdir, rab); //Console.Read();
Process.Start("Vomand.cmd");
Directory.Move(@"c:\v\1.exe", subdir); //Console.Read();
}
}
}
}
}
```

Работа этой программы в автоматическом режиме на экране не отображается. Добавим в программу отладочные операторы, останавливающие её в контрольных точках. После остановки средствами операционной системы проанализируем состояние необходимых файлов и каталогов.

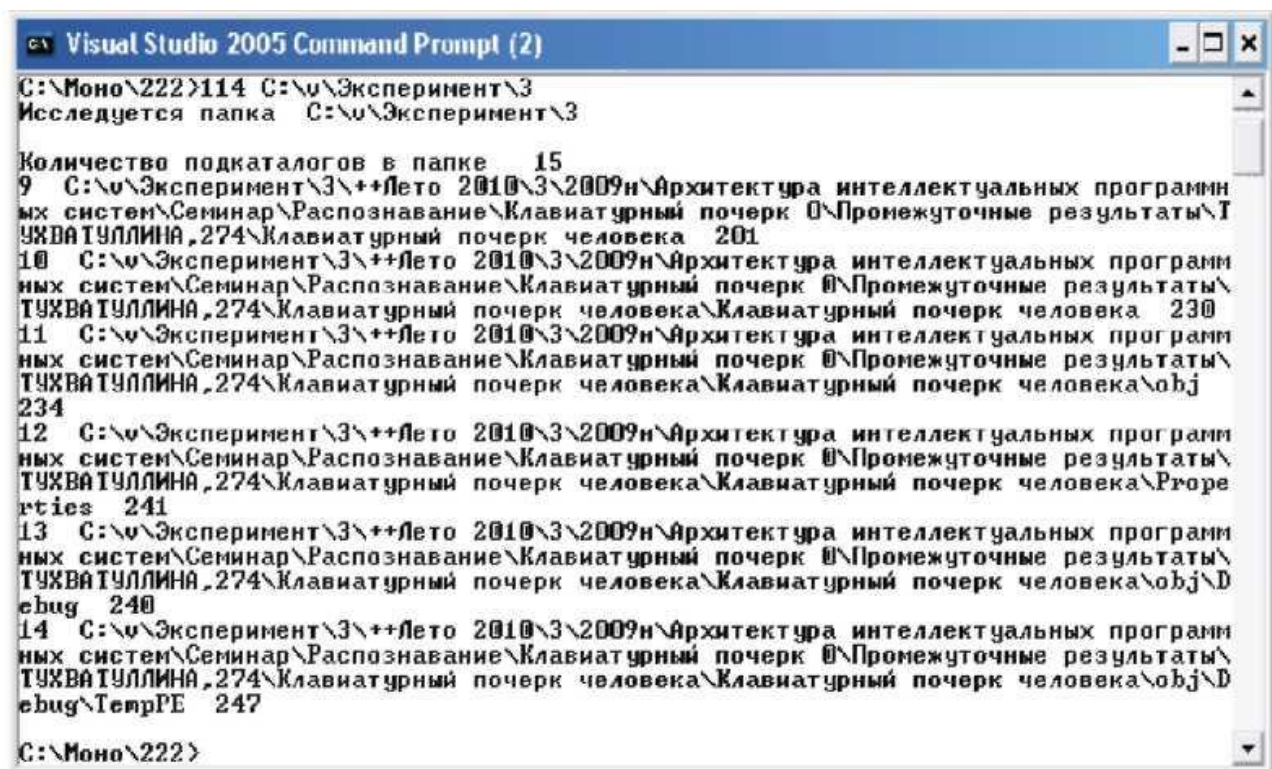
Фрагмент содержания папки `c:\v` перед началом работы программы `remont2.exe`:



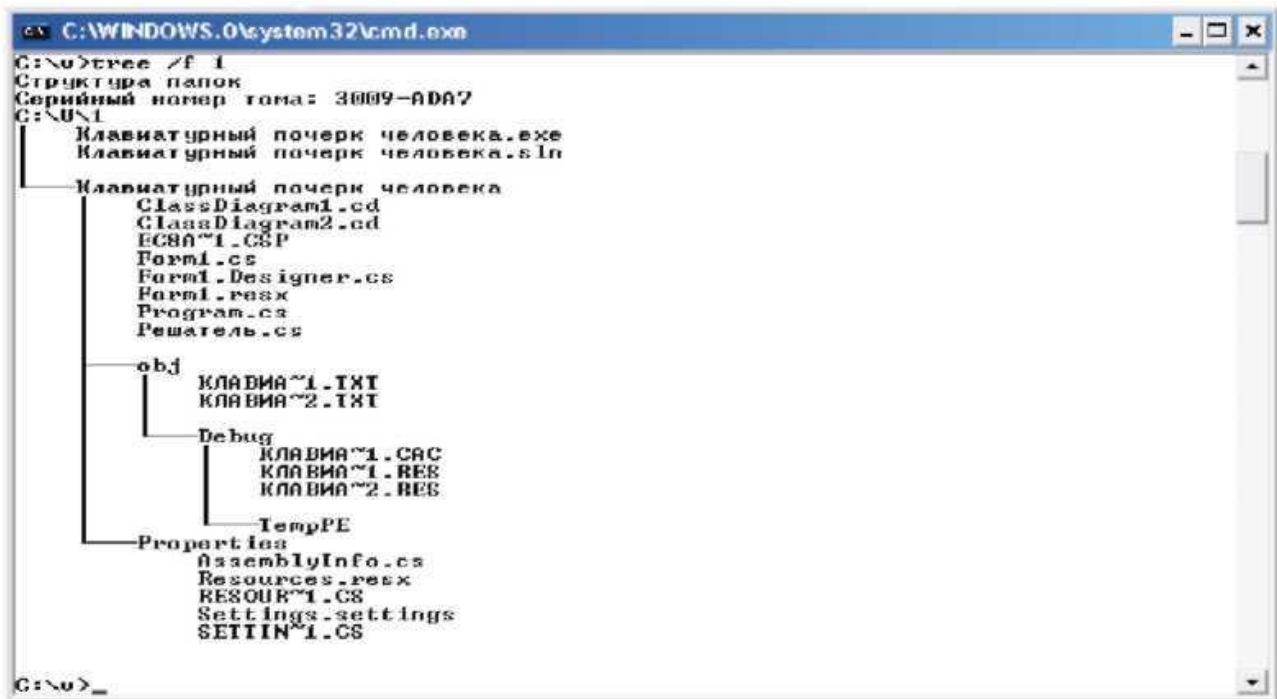
Текст командного файла comand.cmd.



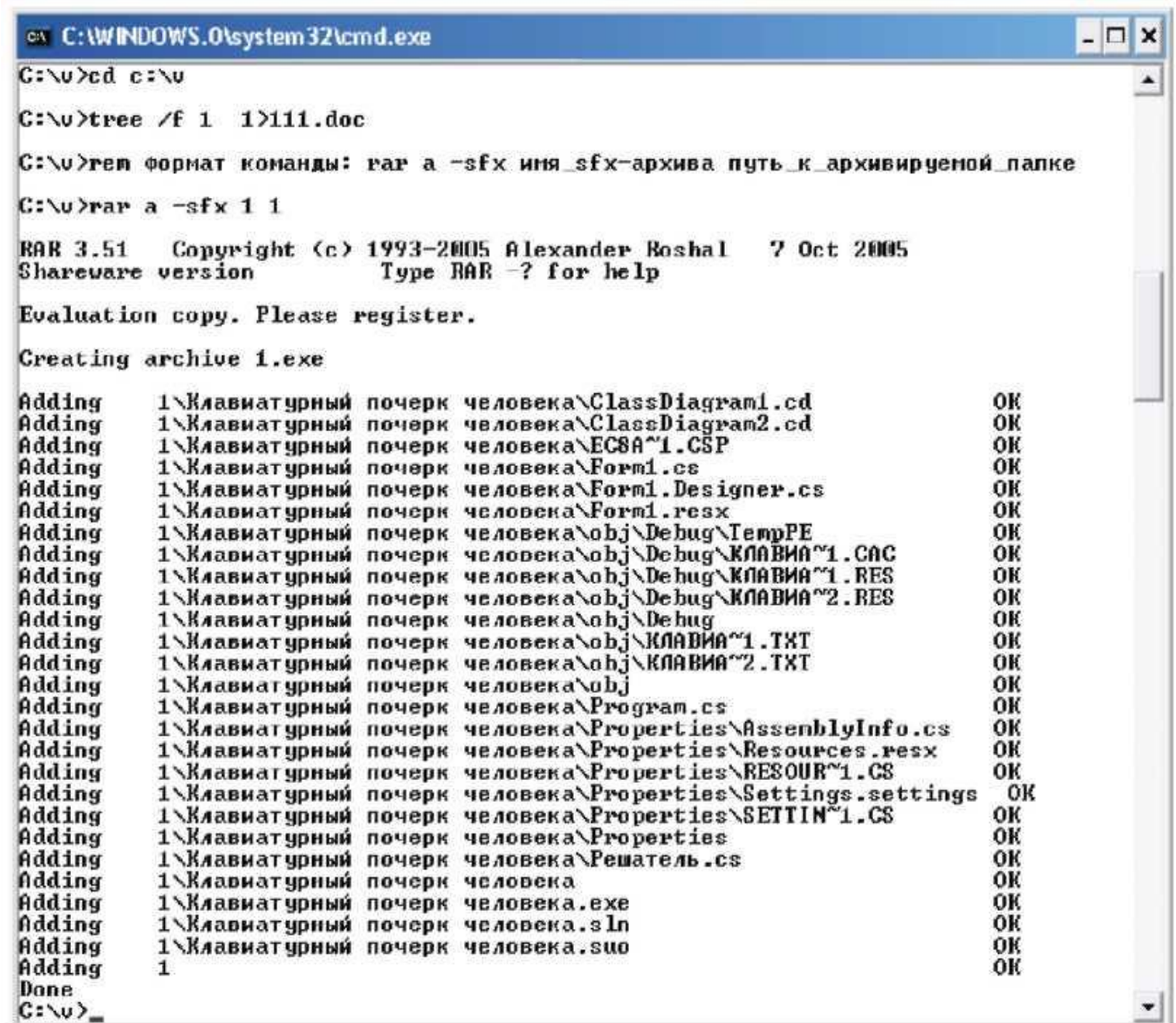
Результат анализа исследуемой папки программой 114.exe перед началом работы программы remont2.exe:



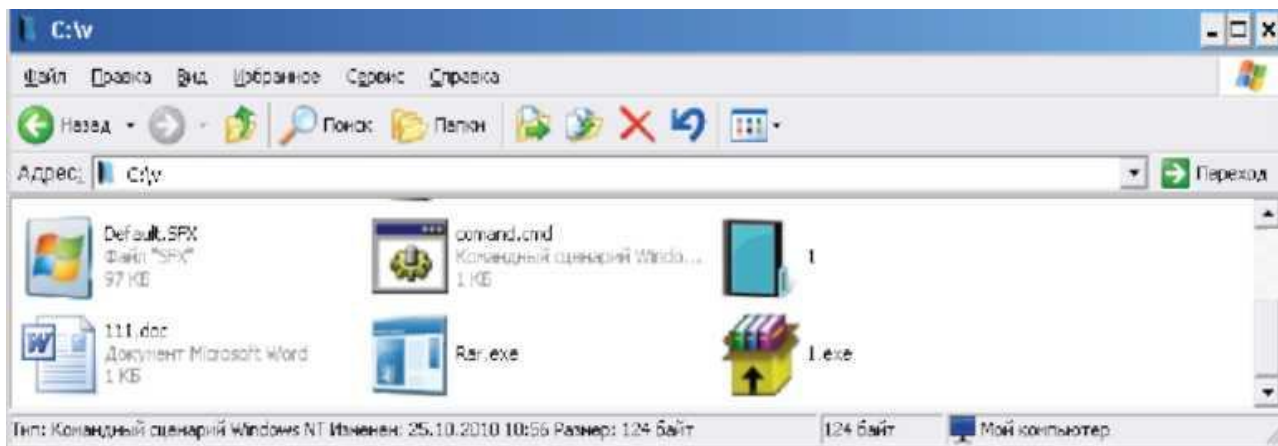
Структура исследуемой папки, полученная перед ремонтом по команде tree операционной системы:



Исполнение командного файла comand.cmd:



Содержимое диска v после работы архиватора:



Из рисунка видно, что на диске добавились папка "1" и sfx-архив 1.exe.

Проверка исследуемого каталога после преобразования по программе 114.exe:



Общее количество каталогов в папке уменьшилось, так как каталоги 10-14 являются подкаталогами каталога 9, имеющего размер имени больше 200 символов. Все они включены в архив 1.exe, что видно из протокола работы командного файла.

СПРАВОЧНЫЕ ДАННЫЕ НЕКОТОРЫХ КОМАНД

Кириченко Александр Аполлонович

COMP

Сравнение содержимого двух файлов или двух наборов файлов.

COMP [данные1] [данные2] [/D] [/A] [/L] [/N=число] [/C]

данные1 Первый файл или набор файлов для сравнения.

данные2 Второй файл или набор файлов для сравнения.

/D Вывод различий в десятичном формате

/A Вывод различий в формате ASCII.

/L Вывод номеров строк, в которых обнаружены различия.

/N=число Число первых строк в каждом файле, которые следует сравнить.

/C Сравнение файлов без учета регистра букв ASCII.

/OFF[LINE] Не пропускать файлы с установленным атрибутом "Автономный".

Для указания наборов файлов в первых двух параметрах используются подстановочные знаки.

DISKCOMP

Сравнение содержимого двух гибких дисков

DISKCOMP [диск1: [диск2:]]

DISKCOPY

Копирование содержимого одного гибкого диска на другой.

DISKCOPY [диск1: [диск2:]] [/V]

/V Копирование с проверкой правильности переноса данных.

Оба диска должны иметь одинаковый тип. Параметры диск1 и диск2 могут ссылаться на одно и то же устройство.

FC

Сравнение двух файлов или двух наборов файлов и вывод различий между ними.

FC [A] [C] [L] [/LBn] [/N] [/OFF[LINE]] [/T] [/U] [/W] [/nnnn][диск1:][путь1] имя_файла1 [диск2:][путь2] имя_файла2 *FC /B [диск1:][путь1] имя_файла1 [диск2:][путь2] имя_файла2*

/A Вывод только первой и последней строк для каждой группы различий.

/B Сравнение двоичных файлов

/C Сравнение без учета регистра символов .

/L Сравнение файлов в формате ASCII.

/LBn Максимальное число несоответствий для заданного числа строк.

/N Вывод номеров строк при сравнении текстовых файлов ASCII.

/OFF[LINE] Не пропускать файлы с установленным атрибутом "Автономный".

/T Символы табуляции не заменяются эквивалентным числом пробелов .

/U Сравнение файлов в формате UNICODE.

/W Пропуск пробелов и символов табуляции при сравнении.

/nnnn Число последовательных совпадающих строк, которое должно встретиться после группы несовпадающих.

[диск1:][путь1] имя_файла1 Указывает первый файл или набор файлов для сравнения.

[диск2:][путь2] имя_файла2 Указывает второй файл или набор файлов для сравнения.

FIND

Поиск текстовой строки в одном или нескольких файлах.

FIND [V] [C] [/N] [/I] [/OFF[LINE]] " строка" [[диск:][путь] имя_файла[...]]

/V Вывод всех строк , НЕ содержащих заданную строку.

/C Вывод только общего числа строк, содержащих заданную строку.

/N Вывод номеров отображаемых строк.

/OFF[LINE] Не пропускать файлы с установленным атрибутом "Автономный".

/I Поиск без учета регистра символов .

"строка" Искомая строка.

[диск:][путь] имя_файла Один или несколько файлов, в которых выполняется поиск.

Если путь не задан, поиск выполняется в тексте, введенном с клавиатуры либо переданном по конвейеру другой командой.

FINDSTR

Поиск строк в файлах.

FINDSTR [/B] [/E] [/L] [/R] [/S] [/I] [/X] [/V] [/N] [/M] [/O] [/P] [/F: файл] [/C: строка] [/G: файл] [/D: список_папок] [/A: цвета] [/OFF[LINE]] строки [[диск:][путь] имя_файла[...]]

/B Искать образец только в началах строк.

/E Искать образец только в конце строк.

/L Поиск строк дословно

/R Поиск строк как регулярных выражений.

/S Поиск файлов в текущей папке и всех ее подпапках.

/I Определяет, что поиск будет вестись без учета регистра.

/X Печатает строки, которые совпадают точно.

/V Печатает строки, не содержащие совпадений с искомыми.

/N Печатает номер строки, в которой найдено совпадение, и ее содержимое.

/M Печатает только имя файла, в которой найдено совпадение.

/O Печатает найденный строки через пустую строку.

/P Пропускает строки, содержащие непечатаемые символы.

/OFF[LINE] Не пропускает файлы с установленным атрибутом "Автономный".

/A:цвета Две шестнадцатеричные цифры - атрибуты цвета. См. "COLOR /?"

/F: файл Читает список файлов из заданного файла (/ для консоли).

/C: строка Использует заданную строку как искомую фразу поиска.

/G: файл Получение строк из заданного файла (/ для консоли).

/D: список_папок Поиск в списке папок (разделяются точкой с запятой).

строка Искомый текст.

[диск:][путь] имя_файла Задает имя файла или файлов.

Использовать пробелы для разделения нескольких искомых строк, если аргумент не имеет префикса /C. Например, 'FINDSTR " Привет мир" a.b' ищет " Привет" или " мир" в файле a.b, а

команда 'FINDSTR /C:" Привет мир" a.b' ищет строку " Привет мир" в файле a.b.

Краткая сводка по синтаксису регулярных выражений:

. Любой символ.

* Повтор: ноль или более вхождений предыдущего символа или класса

a Позиция в строке: начало строки

\$ Позиция в строке: конец строки

[класс] Класс символов : любой единичный символ из множества

[Акласс] Обратный класс символов: любой единичный символ из дополнения

[x-y] Диапазон: любые символы из указанного диапазона

\x Служебный символ: символьное обозначение служебного символа x

\<хуz Позиция в слове: в начале слова

хуz\> Позиция в слове: в конце слова

За полной информацией о регулярных выражениях FINDSTR обратитесь к доступной интерактивной документации.

PUSHD

Сохранение значения текущей активной папки и переход к другой папке.

PUSHD [путь | ..]

путь Каталог, который следует назначить текущим.

Когда расширенная обработка команд включена, команда PUSHD допускает ввод сетевых путей в дополнение к обычным именам дисков и путям.

Если указан сетевой путь, команда PUSHD создает временное имя диска, указывающее на заданный сетевой ресурс, а затем производит смену текущего диска и каталога, используя вновь определенное имя диска.

Выделение временных имен дисков проводится в обратном порядке начиная с Z:, причем выбирается первое свободное имя диска.

POPD

Восстановление предыдущего значения текущей активной папки, сохраненного с помощью команды PUSHD. Когда расширенная обработка команд включена, команда POPD удаляет временные имена дисков, созданные командой PUSHD для сетевых ресурсов, при удалении соответствующего диска из стека каталогов.

PRINT

Вывод на печать содержимого текстовых файлов.

PRINT [/D:устройство] [[диск:][путь] имя_файла[...]]

/D:устройство Устройство для печати.

SORT

Сортировка ввода.

SORT [/R] [/+n] [/M килобайтов] [/L язык] [/REC символов] [[диск1:][путь1] имя_файла1] [/T [диск2:][путь2]] [/O [диск3:][путь3]имя файла3] /+n

Задаёт число символов, n, до начала каждого сравнения. /+3 показывает, что каждое сравнение будет начинаться с третьего символа каждой строки. Строки меньше чем n символов собираются перед всеми остальными строками.

По умолчанию, сравнение начинается с первого символа каждой строки.

/L[OCALE] язык Перекрывает установленные в системе по умолчанию язык и раскладку заданными. Пока существует возможность только одного выбора: ""С"" - наиболее быстрый способ упорядочивания последовательности. Сортировка всегда идет без учета регистра.

/M[EMORY] килобайтов Задаёт количество основной памяти, используемой для сортировки, в килобайтах. Размер памяти должен быть не менее 160КБ. При явном задании размера памяти именно это количество будет, использовано невзирая на то, какое количество основной памяти доступно в системе. Наилучшей производительности можно добиться, не задавая размер памяти. По умолчанию, сортировка выполняется за один проход (без временного файла), используя максимально доступный размер памяти. В остальных случаях, сортировка выполняется за два прохода (с сохранением частично отсортированных данных во временном файле), так что количество памяти, используемой для обоих проходов, - одинаково. По умолчанию, максимальный объем памяти равен 90% доступной основной памяти, если входными и выходными потоками являются файлы на диске, и 4 5% доступной основной памяти - иначе .

/REC[ORD MAXIMUM] символов Определяет максимальное число символов в записи (по умолчанию 4096, максимальное 65535).

/R[EVERSE] Обратный порядок сортировки; т.е. сортировка идет от Я до А, и затем от 9 до 0.

[диск1:][путь1]имя файла1 Определяет имя сортируемого файла. Если оно опущено, то будет использоваться стандартный поток ввода. Явное задание сортируемого файла работает быстрее, чем перенаправление того же файла в качестве стандартного потока ввода.

/T[EMPORARY] [диск2:][путь2] Определяет путь к папке, содержащей рабочие файлы сортировки, в том случае, когда данные не помещаются в основной памяти. По умолчанию используется системная временная папка.

/O[UTPUT] [диск3:][путь3] имя_файла3 Определяет имя файла, в котором сохраняются

отсортированные результаты. Если оно опущено данные записываются в стандартный поток вывода .

Явное задание файла вывода работает быстрее чем перенаправление стандартного потока вывода в этот же файл.

TREE

Графическое отображение структуры папок заданного диска или заданной папки.

TREE [диск :][нуть] [/F] [/A]

/F Вывод имен файлов в каждой папке. */A* Использовать символы ASCII вместо символов национальных алфавитов.

VERIFY

Установка режима проверки правильности записи файлов на диск

VERIFY [ON | OFF]

Введите VERIFY без параметра для определения текущего значения этой команды.

VOL

Вывод метки и серийного номера тома для диска.

VOL [диск:]

ЭКОНОМИЧЕСКИЕ НАУКИ

ЭКОНОМИЧЕСКАЯ СОЦИОЛОГИЯ КАК НАУКА

Шайтова Нургуль Жеткергеновна

Экономическая социология - это междисциплинарная наука, потребность в которой вызвана наличием проблем, не доступных для решения дисциплинарно ограниченным знанием.

Предметом экономической социологии являются законы, категории, методы, которые формулируются по принципу взаимодополнения. Основой такого взаимодополнения должна стать определенная парадигма. Как отмечалось, в западной экономической социологии этому способствовали методология теории организации и концепция социального действия.

Исследователь Г. Бергер, анализируя взаимодействие наук, различает такие его виды: мультидисциплинарный, плюрадисциплинарный и трансдисциплинарный. Каждый из этих видов предполагает определенные специфические способы объединения научных дисциплин начиная с простого обмена идеями и заканчивая синтезом целых концепций, методологий, процедур, терминов, а также исследовательской деятельности.

Экономической социологии часто приписывается статус специальной социологической теории. Такого мнения придерживаются социолог Р. Штольберг, известный систематизатор и историк социологической науки Дж. Тернер и академик Т. Заславская. По мнению Т. Заславской, экономическая социология является одной из сорока ветвей отраслевых социологических теорий и ее задача заключается в воспроизведении целостной картины функционирования экономики в системе социального процесса. Поэтому следует различать четыре группы проблем:

- разработку экономико-социологической модели определенного состояния общества;
- обоснование социальной стратегии и тактики государства с учетом интересов различных социальных групп;
- социолого-экономическое изучение социальной структуры населения как субъекта экономической деятельности;
- системное исследование социальных регуляторов экономики, конкретных механизмов влияния на нее социальных отношений.

Социологический подход к обществу позволяет построить модель как целостной общественной системы, так и отдельных ее подсистем, в частности социальной сферы. Социологическая наука включает в себя категории и обобщения, с помощью которых можно описывать и определять закономерности функционирования этой области жизни, ее связи с экономикой, политикой и культурой. Деятельность государства, называемая социальной политикой, представляет собой комплекс механизмов, призванных обеспечить следующие направления жизнедеятельности граждан: повышение общественного благосостояния; улучшение качества жизни народа; обеспечение социальной защиты и политической стабильности. Таким образом, социальная политика имеет три главные функции: экономическую (стимулирование различных видов экономической деятельности), политическую (создание социально-экономических предпосылок для сохранения политического равновесия в обществе) и стратегическую (создание социально-экономических условий для сохранения и развития "человеческого капитала", обеспечение необходимого уровня жизни для всех групп населения, доступа к медицинскому обслуживанию, образованию, культурным ценностям).

Современные социологи Б. Дикон, М. Хале и П. Стабс в работе "Глобальная социальная политика" высказывают суждения, имеющие важное значение для понимания соотношения

экономических и социальных факторов в системе социальной политики современных развитых государств мира. Эти ученые разрушают примитивное представление о прямой связи между уровнем экономического развития страны и благосостоянием населения, которое, к сожалению, стало расхожим политическим штампом в молодых постсоветских демократиях.

В раннем исследовании сравнительного развития благосостояния стран социологи установили связь между усилиями государства по обеспечению благосостояния народа и экономическим развитием страны. Способ измерения усилий, направленных на повышение благосостояния, посредством определения процента от валового национального продукта (ВНП), расходуемого на социальное обеспечение, вызывал большие сомнения. В результате экономическое развитие было определено как предпосылка социального благосостояния. Вместе с тем, на высшей ступени экономического развития находились, по-видимому, проблематичные случаи, особенно в США, стране, которая, несмотря на дальнейшее экономическое процветание, явно оставалась "отстающим государством благосостояния". Как показал дальнейший сравнительный анализ социальной политики, выше определенного уровня экономических предпосылок не существует необходимой связи между усилиями государства по обеспечению благосостояния общества и его экономическим уровнем. При этом определенные коррективы вносит политический выбор. Характерной особенностью взаимосвязи экономического развития и социального благосостояния стало возрастание имущественного неравенства в периоды экономического подъема. Ортодоксальные экономисты в 70-80-х годах XX в. утверждали, что увеличение неравенства доходов среди населения страны является платой за экономический прогресс: чем более проявляется неравенство в стране, тем она богаче и, как следствие, лучше живется ее гражданам. Эти ортодоксальные идеи в настоящее время подвергаются сомнению.

Социальная политика (цель которой - обеспечить каждому справедливую долю ресурсов) может содействовать экономическому развитию страны. Из доклада ОЕСР (1996) следует неожиданный вывод: будущий расцвет иногда зависит от замедления темпов роста имущественного неравенства граждан.

Набор предлагаемых далее индикаторов предусматривает соотношение главных факторов, корректирующих экономические показатели. Они образуют определенную системную модель, куда входят макроэкономические, социально-экономические и социальные показатели, характеризующие определенное состояние экономики страны, определенный политический режим и зависящую от них социальную политику.

Группу социально-экономических показателей образуют данные, характеризующие уровень безработицы, процент лиц, находящихся за чертой бедности, соотношение средней заработной платы и стоимости потребительской корзины.

К социальным показателям, согласно методике ООН, относится индекс человеческого развития, состоящий из трех компонентов - показателей: продолжительности жизни, уровня образования населения и среднедушевого дохода. Взаимосвязь указанных показателей может приобретать характер определенных моделей функционирования общества. В случае кризисного состояния экономики государство осуществляет политику чрезвычайных мер; стабильному состоянию экономики соответствует политика стимулирования со стороны государства; с экономикой динамического роста связана политика регулирования.

Темпы изменения объема производства могут колебаться от -20 % (экономический кризис) до нуля (стабильность). Если этот показатель достигает +10 %, в стране наблюдается динамическое развитие экономики. Другой экономический показатель - уровень инфляции - свидетельствует о кризисном состоянии страны: если он составляет 50 %, в экономической стабильности, если не превышает 30 %, и о динамическом развитии, если равен 5 %.

На выбор модели политического управления существенно влияет уровень безработицы. В условиях экономического кризиса этот уровень достигает 20 % и влечет за собой политику чрезвычайных мер. Предельным считается уровень безработицы 10 %. Он свидетельствует о стабильной экономике и сопровождается политикой стимулирования. Безработица на уровне 5 % расценивается как естественная. Она наблюдается в условиях экономического развития страны.

СОВРЕМЕННАЯ ЗАПАДНАЯ ЭКОНОМИЧЕСКАЯ СОЦИОЛОГИЯ

Шайтова Нургуль Жеткергеновна

В современной западной науке экономическая социология имеет самостоятельный статус с определенным кругом проблем. В 80-е годы XX в. эта новая теория получила официальное признание, что было отмечено на международном семинаре, состоявшемся в Париже в 1987 г. Ученые констатировали, что в последнее время экономическому анализу подверглось много новых областей: образование, семья, социальное взаимодействие, девиантное поведение, организационный и коллективный выбор. Первые разработки в области экономической социологии принадлежат известному американскому экономисту Г. Беккеру. Именно он предложил объединительную идею, социологическая сущность которой состоит в том, что всем видам социального поведения присущи три признака: оптимизация поведения, рыночная природа и наличие устойчивых предубеждений. За истекшее с тех пор время в экономической социологии появились и своя проблематика, и свои разработчики. Так, проблему выбора исследуют О. Льюис и Гарбуа, проблему потребительского поведения - Ф. Ириберн, влияние политики правительства на динамику потребностей - Г. Телок, поведение семей - Б. Леменишер, трудовое поведение - Г. Жинтис.

Актуальность экономической социологии объясняется, по мнению Г. Беккера, тем, что к традиционным объектам экономики - товар, труд, финансы - добавляются феномены, не имеющие рыночной природы. Это касается сферы услуг и ценностей культуры, т. е. нетоварного сектора. Все, что прежде всего присуще "Экономиксу" (например, распределение дефицитных ресурсов), в той же мере свойственно и сфере политики (распределение власти, налогов, заказов), и повседневной жизни (распределение времени между периодами сна и бодрствования), и научно-исследовательской деятельности (распределение времени и энергии между исследовательскими программами). Эквивалентом в процессе обмена служит полезность независимо от того, идеи ли это или товары. Такой же является и сущность цены как меры полезности вещи, потребности в ней, расчета за пользование ею.

В западной науке объединение социологии и экономики происходит не только в рамках социологии. Основатели новой междисциплинарной области опираются на определенную экономическую концепцию, согласно которой в центре "Экономикса" оказываются законы обмена, распределения, использования ограниченных ресурсов. Такой взгляд на предмет политической экономии осуждался марксизмом. Предмет политэкономии марксизм усматривал прежде всего в законах производства, в отношениях собственности. К. Маркс критиковал вульгарный социализм, который позаимствовал у буржуазных экономистов способ рассмотрения и трактовки распределения как нечто независимое от производства.

Какие же методологические преимущества имеет парадигма распределения и обмена? Прежде всего это то, что ее авторы замечают главное противоречие, которое нарушает равновесие экономической системы, а именно периодическое рассогласование между производством и потреблением.

Практическая направленность социоэкономических разработок также отличает современные научные исследования от предшествующих, когда старались решать преимущественно теоретико-познавательные задачи. Главным недостатком классической науки, по мнению Ч. Арджириса, является дисциплинарная ограниченность, которая мешала научной мысли найти

практический выход в управлении реальными организациями.

Преобразование теории в социальную технологию ученый усматривает в отказе от надличностных категорий “товар”, “стоимость”, “деньги” и в обращении к поведенческим концепциям.

Социологи Мак-Клеланд, Дж. Эткин и Дж. Катона продемонстрировали новые возможности синтезированного знания. В основе их подхода лежит социология ожиданий “ожиданий”, а центральной категорией становится (achievement) “достижение”. В предложенной теории проявляется влияние психологических школ У. Джемса и К. Левина. Личностная характеристика - стремление к достижению - становится главным мотивом экономического процветания общества или нации. Заложенные с детства черты: ощущение индивидуального успеха, умение преодолевать трудности, удовлетворение от достижений становятся основными побудительными мотивами предприимчивости. Социологи Дж. Эткин и А. Шаррон полагают, что мера свободного развития общества зависит от “синдрома достижения”, которым обладают его члены, “это - динамичный потенциал всех капиталистических наций”.

Принципы новой науки в значительной мере реализовал другой представитель экономической социологии - Дж. Катона. Его считают революционером в области экономики. Заслуга Дж. Катона состоит в том, что он обратил внимание на сферу потребления, которую представил в виде системы поведения. Основные отношения в ней разворачиваются не вокруг инвестиций, производства и потребления, а вокруг личности потребителя. Такой аспект позволяет сконцентрироваться прежде всего на значении психологических проявлений, в частности на социально-психологических свойствах различных социальных групп, к которым относится индивид. Поведение потребителя - глобальный фактор экономики, влияющий на прогнозы рыночной ситуации. Рыночные отношения (цена, спрос и предложение) изменяются в зависимости от выбора потребителем товаров и услуг, а также от того, собирается ли он вкладывать свои накопления, хранить их или что-то приобретать. Решения обуславливаются прежде всего особенностями психологии личности, ее ожиданиями, ценностными ориентациями, т.е. тем, что психологи и социологи называют преддиспозицией. Отсюда становится возможным переход к методологии изучения прогнозирования и поведения потребителя. Предметом изучения становятся инструментальные и базовые ценности социальных групп. Знание их позволяет построить трендовые модели потребительского поведения, проследить главные тенденции динамики спроса. Дж Катона предлагает модель поведения потребителя, в которой учитывает состояние окружающей среды с его составляющими: уровнем инфляции, политической ситуацией, ролью большого бизнеса, его давлением на спрос. Ему принадлежит схема-формула модели потребительского поведения.

Изучение феномена потребителя имеет особое социальное значение еще и потому, что ему противостоят организованный бизнес или государственный монополист. Потребитель нуждается в защите со стороны государства, что имеет место в развитых странах, где в структуру государственного управления хозяйством обязательно входят специальные департаменты. Например, в США федеральные министерства, отделы и службы контролируют производственную деятельность предпринимателей, предостерегая их от нарушений интересов потребителя. Этой задаче соответствуют стандарты Комиссии профессиональной безопасности и здоровья. Стандарты отделов производства должны учитывать требования агентств по защите окружающей среды. Продукция подпадает под стандарты администрации продовольствия и лекарств, министерства здравоохранения, образования, социальной защиты. Такие же функции выполняют комитеты по защите продуктов потребления, отделы маркировки, рекламы, транспорта. Совет по заработной плате и стабильности цен контролирует повышение цен большими национальными фирмами.

Следовательно, сегодня бизнес и рынок не могут существовать без государственного регулирования и контроля общественности, с одной стороны, и учета интересов потребителя - с другой. Одна из норм современного бизнеса гласит: "Бизнесмен должен учиться думать, обсуждать и действовать на более широкой основе, нежели только экономические расчеты". Современная наука о бизнесе уже не ограничивается экономическим сознанием, ориентированным на благосостояние. Она все более тяготеет к социологическим и психологическим теориям мотивированного экономического выбора.

Отсутствие на потребительском рынке товаров, которые входят в постоянный рацион или ассортимент определенной нации, народа, сложность их приобретения и представляют собой "относительное обнищание". Это важное замечание должно напоминать "чистым" экономистам, что экономика не существует в социальном вакууме. Лишь культурная среда (традиции, обычаи, ценности, символы, значения, смыслы) делает вещь товаром. Человек не довольствуется любой одеждой только ради того, чтобы она его согревала. Одежда приобретает особый смысл, поскольку символизирует социальный статус личности, сферу деятельности и пр. Образ потребления зависит от принадлежности индивида к определенной социальной группе. Ценностно-культурный контекст превращает вещь в символ престижа. По мнению Ф. Ириберна, "владение престижной вещью предоставляет обладателю ощущение преимущества".

Влияние феноменологической социологии на экономическую науку сказалось также и на исследовательских приемах. Увлечение методом микроанализа в социэкономике в значительной мере стало результатом распространения качественных методов. Их "философия" требует, чтобы исследователь "погружался" в ежедневное бытие человека, изучая его мир, "схватывая чистые формы сознания", в частности стереотипы, порожденные ценностями возрастных, этнических или территориальных групп. Например, американские специалисты по микроанализу из университета штата Висконсин и Массачусетского технологического института У. Фокст, Т. Гринберг, Дж. Кембелл и А. Ривлин изучают корреляции между принадлежностью личности к определенной социальной группе, ее экономическим состоянием и выбором определенного типа действия, направленного на приобретение вещей, сохранение или вложение денег, поиск работы, бракосочетание и т. п. Это - вероятностное исследование вероятностных прогнозов поведения - характеризуют свой метод авторы, не содержащее философского понимания каузальности. Его сущность состоит в получении информации относительно отдельных элементов организации, например одного домашнего хозяйства. Если предвидение оправдано в конкретном случае (на примере определенной семьи), то его можно экстраполировать и на совокупность домашних хозяйств. Отказываясь от искусственных экспериментов, приверженцы микроанализа считают необходимым находить интересующие их факты в самой жизни. Дж. Кембелл назвал этот метод квазиэкспериментом.

Итак, для современной экономической социологии характерны:

- психосоциологическая редукция экономической науки;
- переход на уровень поведенческих теорий;
- использование категорий, позволяющих описать содержание, мотивы экономического поведения;
- учет влияния неэкономических факторов на экономическое поведение;
- прогнозирование реакции групп населения на экономические инновации;
- изучение влияния массового сознания и массового поведения на рыночные изменения (динамику цен, потребительскую ситуацию и спрос).

ПРИМЕНЕНИЕ ТЕОРИИ ХАОСА К АНАЛИЗУ РЫНКОВ

Чефранова Ольга Игоревна

Теория хаоса основывается на положении о бессмысленности попыток постичь рыночные закономерности методами математики или статистики и отсутствии на рынке каких-либо выраженных циклических процессов [<http://habrahabr.ru>]. Согласно теории хаоса, рынки являются нелинейными, развивающимися во времени системами. Эта теория является математическим аппаратом для описания поведения подобных систем. Исследования рынка с помощью этого аппарата показывают, что изменение рыночных цен, в основном, случайно, а трендовая компонента – весьма невелика. Доля трендовой компоненты различна для разных рынков и временных интервалов.

Для описания случайных процессов на рынках используют понятие фракталов. Фракталом называют объект, имеющий свойство самоподобия или объект, любая часть которого, подобна целому. Это как ветви дерева – хотя по мере удаления от ствола они становятся все меньше и тоньше, но структуру целого дерева, при этом, сохраняют полностью. Таким же свойством обладает и движение цен на недельных, дневных, часовых и т.д. графиках. Можно видеть все больше мелких подробностей на графиках, но общий характер и направление движения сохраняются.

К числу свойств хаотичных рынков относят также «чувствительность к начальным условиям». Это именно тот параметр, который обуславливает трудность прогнозирования поведения динамичных рынков. Невозможно точно учесть все параметры текущей ситуации (начальные условия), а ошибки, возникшие по этой причине, с течением времени многократно растут из-за сложности системы. Таким образом, точный прогноз поведения системы становится невозможным.

Многие трейдеры полагают, что торговать на 5-минутных барах, например, значит торговать на случайном шуме, а это пустая трата времени. Считается, что торгующие на шуме, имеют весьма небольшую прибыль и со временем потерпят неудачу, т.к. за саму торговлю тоже нужно платить (накладные, комиссия и пр.). С другой стороны, те же трейдеры уверены, что движения цен на больших временных отрезках – не случайны. Поэтому, двигаясь за трендом, можно вполне успешно торговать на недельных или дневных графиках.

Возникает вопрос – почему один и тот же рынок на малых временных интервалах абсолютно непредсказуем, а на состоящих из множества малых, больших интервалах, поведение этого рынка закономерно и предсказуемо?

В реальности это несоответствие – кажущееся. В соответствии с законами математической статистики, система может вести себя случайным образом на малых временных отрезках, и вполне предопределенно – на больших.

На рынке отсутствуют краткосрочные паттерны и циклы, сколько-нибудь значимые с точки зрения их использования в практической торговле. Паттерны ценовых графиков и индикаторов несложно обнаружить в любой последовательности случайных чисел.

Торгуя с учетом долгосрочной рыночной тенденции (долгосрочного тренда), трейдер получает статистическое преимущество. В этом собственно и заключается принцип работы тренд-следящих систем [<http://www.forexcenter.ru>]. И в этом состоит секрет того, почему хорошие тренд-следящие системы на протяжении многих лет функционируют прибыльно, а трейдеры,

использующие внутрисудневную торговлю, на длительных временных промежутках убыточны.

На основании опыта трейдеров, можно сказать, что третья часть успеха заключена в торговой системе, еще треть зависит от выбранного рынка, и последняя треть зависит от дисциплины трейдера в отношении следования выбранной системе. Невозможно предсказать в какой момент торговая система реализует свое статистическое преимущество. Можно только разработать ее и протестировать, не основываясь на исторических данных. Тестировать систему на исторических данных бессмысленно, и, более того, вредно.

Избежать подгонки системы под исторические данные можно, используя единые правила для любых рынков. Тестировать систему следует на максимальном числе доступных рынков. Если длительное время на большинстве рынков система прибыльна, то, вероятно, она не подогнана под исторические данные.

Немаловажным фактором в управлении риском является соблюдение оптимального соотношения размера счета и вероятных потерь. Вероятность потерь тесно связана с прибыльностью, поэтому для снижения потерь следует ограничить прибыльность. Нельзя оценить оптимальный состав вашего портфеля, пока вы не оцените долговременные потери при торговле по всем рынкам, входящим в портфель. Соответственно, начальный размер портфеля не должен быть меньше суммы удвоенных максимальных потерь и маржи.

УЧЕТ КАК СОВРЕМЕННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА

Шайтова Нургуль Жеткергеновна

В настоящее время, когда организациям дана самостоятельность в разработке производственных программ, планов производственного и социального развития, в определении стратегии в области ценовой политики, существенно возрастает ответственность руководителей за принимаемые управленческие решения. Для выработки эффективных и оперативных решений управляющим нужна информация о производственном и финансовом положении организации. Решением второй части этой задачи занимается бухгалтерская служба. В самом общем виде бухгалтерский учет – это информационная система, которая измеряет, обрабатывает и передает по требованию пользователей финансовые данные. В условиях рыночной экономики значительно усложнился процесс управления организацией. Следовательно, усложняются и задачи, стоящие перед системой бухгалтерского учета. Весь бухгалтерский учет делится на финансовый, налоговый и управленческий. В финансовом учете формируется информация о доходах и расходах организации, размерах дебиторской и кредиторской задолженности, величине финансовых инвестиций и доходов от них, состоянии источников финансирования и т.д., которая необходима для составления бухгалтерской (финансовой) отчетности.

Налоговый учет также охватывает информацию, которая не только используется внутри организации, но и сообщается внешним пользователям для проверки достоверности формирования налоговой базы. Для налогового учета является характерным учет доходов и расходов организации согласно Налоговому кодексу РФ, в связи с чем бухгалтерская прибыль и прибыль для налогообложения будут иметь существенные различия. Отметим также, что финансовое состояние предприятия не характеризуется данными налогового учета, так как его назначение ограничивается определением налоговой базы по тому или иному налогу.

Управленческий учет охватывает все виды учетной информации, необходимой для управления в пределах самого хозяйствующего субъекта. Основной составляющей управленческого учета является производственный учет, под которым обычно понимают учет издержек производства и анализ полученной экономии или перерасхода по сравнению с предыдущими периодами, прогнозными значениями или нормативами. Основная цель управленческого учета – обеспечение текущей аналитической и прогнозной информацией руководства и менеджеров, ответственных за достижение конкретных экономических показателей. Процесс подготовки такой информации может существенно отличаться от того, что используется в финансовом учете. Вместе с тем результаты, получаемые на основании данных финансового учета, должны соответствовать результатам, получаемым в управленческом учете, хотя и не будут идентичны им.

На выходе бухгалтерской информационной системы формируются отчеты для:

- внутренних и внешних пользователей бухгалтерской информации;
- целей периодического планирования (прогнозирования), контроля и анализа;
- принятия решений в нестандартных ситуациях и при выборе политики организации.

Бухгалтерия находится на стыке информационных потоков разных подразделений, и практически только она может формировать информацию о реальном финансовом состоянии

предприятия. Следовательно, нередко именно от того, владеет ли руководство оперативной и достоверной информацией, полученной от бухгалтерских служб, зависит, будет ли предприятие банкротом либо будет развиваться стабильно и динамично. Круг пользователей содержащейся в финансовых документах информации включает различные категории — от серьезных аналитиков до случайных “любителей”. Все они пользуются информацией об организации, но с разной степенью понимания и компетенции. Различают внутренних и внешних пользователей бухгалтерской отчетности.

Внешние пользователи в свою очередь делятся на две группы:

1. непосредственно заинтересованные в деятельности организации;
2. опосредованно заинтересованные в ней.

К первой группе внешних пользователей относятся следующие:

1. государство, прежде всего в лице налоговых органов, которые проверяют правильность составления отчетных документов, расчета налогов, определяют налоговую политику;
2. существующие и потенциальные кредиторы, использующие отчетность для оценки целесообразности предоставления или продления кредита, определения условий кредитования, усиления гарантий возврата кредита, оценки доверия к организации как к клиенту;
3. поставщики и покупатели, определяющие надежность деловых связей с данным клиентом;
4. существующие и потенциальные собственники средств организации, которым необходимо определить увеличение или уменьшение доли собственных средств и оценить эффективность использования ресурсов руководством организации;
5. служащие, интересующиеся данными отчетности с точки зрения уровня их заработной платы и перспектив работы в организации.

Вторая группа внешних пользователей бухгалтерской отчетности - это те, которым изучение отчетности необходимо для того, чтобы защитить интересы первой группы пользователей отчетности. К этой группе относятся:

1. аудиторские службы, проверяющие соответствие данных отчетности установленным правилам с целью защиты интересов инвесторов;
2. консультанты по финансовым вопросам, использующие отчетность в целях выработки рекомендаций своим клиентам относительно помещения их капиталов в ту или иную компанию;
3. биржи ценных бумаг, оценивающие информацию, представленную в отчетности, при регистрации соответствующих фирм, принимающие решения о приостановке деятельности какой-либо компании, дающие заключение о необходимости изменения методов учета и составления отчетности;
4. законодательные органы;
5. юристы, нуждающиеся в отчетной информации для оценки выполнения условий контрактов, соблюдения законодательных норм при распределении прибыли и выплате дивидендов, а также для определения условий пенсионного обеспечения;
6. пресса и информационные агентства, которым отчетность необходима для подготовки обзоров, рассмотрения деятельности отдельных компаний и отраслей, расчета обобщающих показателей финансовой деятельности;
7. государственные организации по статистике, использующие отчетность для статистических обобщений по отраслям и сравнительного анализа результатов деятельности на отраслевом уровне;
8. профсоюзы, заинтересованные в отчетной информации для определения своих требований в отношении заработной платы и условий трудовых соглашений, а также для оценки

тенденций развития отрасли, к которой относится организация.

К внутренним пользователям отчетности относятся:

- высшее руководство организации;
- общее собрание участников;
- работники - с точки зрения перспектив работы в организации;
- управляющие соответствующих уровней, которые по данным отчетности определяют правильность принятых инвестиционных решений, эффективность структуры капитала, основные направления политики дивидендов, составляют прогнозные формы отчетности и осуществляют предварительные расчеты финансовых показателей предстоящих отчетных периодов, оценивают возможности слияния с другой организацией или ее приобретения, структурной реорганизации.

Недостатком существующей системы бухгалтерского учета как основного поставщика экономической информации следует признать слабую привязку его к конкретным отраслевым условиям и стилю управления. Такое положение обусловлено тем, что учет организуется в соответствии с нормативными документами общего характера. Учет на предприятиях организован в отрыве от целевых установок управленческой системы. Если признать верным положение о том, что учет должен организовываться с ориентацией на потребности управления в информации, то система бухгалтерского учета должна зависеть от организационной структуры управления. Руководство нередко считает, что бухгалтерия знает все и потому от нее можно требовать любую справку, любую информацию, забывая о том, что на самом деле в бухгалтерии эта информация должна трансформироваться, прежде всего, под требования внешней финансовой отчетности. Возникает перегрузка бухгалтерии не свойственной ей работой. С этой проблемой можно бороться двумя способами. Первый - оптимизация структуры системы управления, второй - применение современных информационных технологий. И эти способы во многом являются взаимосвязанными и взаимодополняющими. Из-за небрежности в бухгалтерском учете предприятие может потерпеть крах. Страдают хозяйствующие субъекты также из-за незнания и соответственно невыполнения последних законов и распоряжений. При ведении бухгалтерского учета вручную возможны и простейшие арифметические ошибки.

Автоматизация бухгалтерского учета на предприятии и подготовка финансовой отчетности в налоговые органы в условиях трансформационной экономики России является одной из наиболее важных задач. Существует широкий выбор различных систем автоматизации бухгалтерского учета. Не следует делить их на плохие и хорошие, сильные и слабые. Все они хороши и их возможности находят практическое применение на предприятиях различного размера, профиля и рода деятельности. При автоматизации следует выбрать необходимую систему, исходя из задач и имеющихся ресурсов. Разумеется, компьютер, не позволит упорядочить бухгалтерский учет и повысить его оперативность, увеличить количество получаемой информации, уменьшить число арифметических ошибок, оценить текущее финансовое положение предприятия и его перспективы. Компьютеру необходим опытный и грамотный бухгалтер.

Таким образом, реализация на практике принципов достаточности, необходимости, непротиворечивости, достоверности повысит информационное обеспечение бухгалтерского учета, что в свое время позволит удовлетворять интересы пользователей финансовой информации.

АНТИКРИЗИСНОЕ УПРАВЛЕНИЕ ФИНАНСАМИ ПРИ УГРОЗЕ БАНКРОТСТВА

Шайтова Нургуль Жеткергеновна

Банкротство является кризисным состоянием и его преодоление требует специальных методов финансового управления. Рыночная экономика выработала обширную систему финансовых методов диагностики банкротства и выработала методiku принятия управленческих решений в условиях угрозы банкротства. Эта методика предназначена не только для предприятий, где кризис очевиден и необходимо принимать неотложные меры по стабилизации, а для всех предприятий, работающих в рыночных условиях, поскольку ее особенности таковы, что позволяют выявить на ранней стадии и устранить негативные факторы развития предприятия, наметить пути их устранения.

Основной методикой являются три положения:

- диагностика банкротства;
- выявление факторов, влияющих на кризисное развитие;
- выработка антикризисных механизмов управления финансами.

Переход к рыночным механизмам хозяйствования привел к появлению нового для Казахстана механизма - механизма банкротства.

Условия и порядок признания предприятия банкротом основываются на определенных законодательных процедурах. Так, решение о признании предприятия банкротом может вынести только арбитражный суд. Основанием для возбуждения судебного производства по делу о банкротстве является письменное заявление:

1. кого-либо из кредиторов;
2. самого предприятия-должника;
3. прокурора.

Кредитор может обратиться с заявлением о возбуждении дела о банкротстве предприятия в случае, если оно не в состоянии удовлетворить в течение одного месяца признанные им претензионные требования либо уплатить долг по исполнительным документам. Предприятие-должник может обратиться в арбитражный суд по собственной инициативе в случае его финансовой несостоятельности или угрозы такой несостоятельности. В современной практике большинство дел о банкротстве возбуждается по инициативе коммерческих банков и налоговых инспекций.

С позиций финансового менеджмента возможное наступление банкротства представляет собой кризисное состояние предприятия, при котором оно неспособно осуществлять финансовое обеспечение текущей производственной деятельности. Преодоление такого состояния, диагностируемого как "угроза банкротства", требует разработки специальных методов финансового управления предприятием.

Рыночная экономика выработала обширную систему финансовых методов предварительной диагностики и возможной защиты предприятия от банкротства, которая получила название "системы антикризисного финансового управления". Для реализации этой системы управления в странах с развитой рыночной экономикой готовятся особые специалисты - менеджеры по

антикризисному управлению компанией. Суть этой системы управления состоит в том, что угроза банкротства диагностируется еще на ранних стадиях ее возникновения, что позволяет своевременно привести в действие специальные финансовые механизмы защиты или обосновать необходимость определенных реорганизационных процедур. Если эти механизмы и процедуры в силу несвоевременного или недостаточно эффективного их осуществления не привели к финансовому оздоровлению предприятия, оно стоит перед необходимостью в добровольном или принудительном порядке прекратить свою хозяйственную деятельность и начать ликвидационные процедуры.

Основной целью антикризисного финансового управления является быстрое возобновление платежеспособности и восстановление достаточного уровня финансовой устойчивости предприятия для избежания его банкротства. С учетом этой цели на предприятии разрабатывается специальная политика антикризисного финансового управления при угрозе банкротства. Она может быть сформулирована следующим образом:

Политика антикризисного финансового управления представляет собой часть общей финансовой стратегии предприятия, заключающейся в разработке и использовании системы методов предварительной диагностики угрозы банкротства и механизмов финансового оздоровления предприятия, обеспечивающих его защиту от банкротства.

Реализация политики антикризисного финансового управления при угрозе банкротства предусматривает. Основное содержание политики антикризисного финансового управления предприятием при угрозе банкротства

Периодическое исследование финансового состояния предприятия с целью раннего обнаружения признаков его кризисного развития, вызывающих угрозу банкротства. В этих целях в системе общего анализа финансового состояния предприятия выделяется особая группа объектов наблюдения, формирующая возможное "кризисное поле", реализующее угрозу банкротства. В процессе исследования показателей "кризисного поля" применяются как традиционные, так и специальные методы анализа. Анализ и контроль таких показателей включается в систему мониторинга финансовой деятельности предприятия.

Определение масштабов кризисного состояния предприятия. При обнаружении существенных отклонений от нормального хода финансовой деятельности, определяемого направлениями его финансовой стратегии и системой плановых и нормативных финансовых показателей, выявляются масштабы кризисного состояния предприятия, т.е. его глубина с позиций угрозы банкротства. Такая идентификация масштабов кризисного состояния предприятия позволяет осуществлять соответствующий селективный подход к выбору системы механизмов защиты от возможного банкротства.

Изучение основных факторов, обусловивших (и обуславливающих в предстоящем периоде) кризисное развитие предприятия. Разработка политики антикризисного финансового управления определяет необходимость, предварительной группировки таких факторов по основным определяющим признакам; исследованию степени влияния отдельных факторов на формы и масштабы кризисного финансового развития; прогнозирование развития факторов, оказывающих такое негативное влияние.

Формирование целей и выбор основных механизмов антикризисного финансового управления предприятием при угрозе банкротства. Цели и механизмы антикризисного финансового управления должны соответствовать масштабам кризисного состояния предприятия и учитывать прогноз развития основных факторов, определяющих угрозу банкротства. С учетом этих условий, финансовый менеджмент на данном этапе может быть направлен на реализацию трех принципиальных целей:

1. обеспечение финансового оздоровления предприятия за счет реализации внутренних резервов хозяйственной деятельности;
2. обеспечение финансового оздоровления предприятия за счет внешней помощи и частичной его реорганизации;
3. прекращение хозяйственной деятельности и начало процедуры банкротства (в связи с невозможностью финансового оздоровления предприятия).

ФИЛОЛОГИЧЕСКИЕ НАУКИ

РОЛЬ ИГРЫ НА УРОКАХ АНГЛИЙСКОГО ЯЗЫКА

Сомова Дарья Михайловна

Человеческая игра есть изначально мотивированная осмысленная деятельность, которая включает в себя предметную, теоретическую и душевную деятельность.

Игра является инструментом преподавания, который активизирует мыслительную деятельность учеников, позволяет сделать учебный процесс привлекательным и интересным, заставляет учеников волноваться и переживать. Это мощный стимул к овладению иностранным языком.

Для школьников игра – путь к познанию, так как играя ученики без особых усилий, с удовольствием включаются в процесс обучения, забывая, что идёт урок. С радостью играют ученики всех возрастов (дети, подростки, юноши), особенно если игра принимает форму соревнования, требующего смекалки, быстрой реакции, хорошего знания предмета.

Игра имеет большое значение в воспитании, в обучении и развитии детей как средство психической подготовки к будущим жизненным ситуациям. Она позволяет преподавателю организовать деятельность учащихся, сделать ее более активной, заинтересовать их в изучении предмета и в поиске дополнительных средств для получения информации – то есть создать мотивацию, которой так часто не хватает нашим ученикам. Игра способствует развитию таких качеств как самостоятельность и инициативность.

Таким образом, игра представляет собой ситуативно-вариативное упражнение, где создается возможность для многократного повторения речевого образца в условия, максимально приближенных к реальному речевому общению с присущими ему признаками – эмоциональностью, спонтанностью, целенаправленностью воздействия.

В настоящее время в методической литературе существует достаточно большое количество классификаций, систематизирующих виды учебных игр в соответствии с тем или иным классификационным критерием. В данной статье игры будут разделены в зависимости от их целей.

Фонетические игры

Цели фонетических игр:

- тренировать учащихся в произношении английских звуков;
- научить учащихся громко и отчетливо читать стихотворения;
- формирование навыков фонетического слуха.

Систематичность применения специальных фонетических имитационных упражнений, фонетически направленных игровых заданий способствует формированию основ фонетических навыков.

Среди фонетических игр, используемых на начальном этапе обучения, можно выделить игры-загадки, игры-имитации, игры-соревнования, игры с предметами, игры на внимательность, игры-лото со звуками.

Лексические игры

Цели лексических игр:

- тренировать учащихся в употреблении лексики в ситуациях, приближенных к естественной обстановке;
- активизировать речемыслительную деятельность учащихся;
- развивать речевую реакцию учащихся;
- познакомить учащихся с сочетаемостью слов.

Слова хранятся у нас в памяти ассоциативно-тематическими группами. Поэтому, формируя у учеников базовый словарь, то есть запас «кирпичиков», из которых они будут строить фразы, нам необходимо вводить новые лексические единицы тематическими группами.

В этой связи учебный материал объединяется вокруг основных лексических тем: семья, внешность, одежда, дом, школа, еда, домашние животные, звери, цвет и другие. Чтобы слова и выражения запоминались прочно, их надо очень много раз повторять.

При обучении лексике целесообразно использовать игры на карточках, картинках или кубиках.

Грамматические игры

Цели грамматических игр:

- научить учащихся употреблению речевых образцов, содержащих определенные грамматические трудности;
- создать естественную ситуацию для употребления данного речевого образца;
- развить речевую активность и самостоятельность учащихся

Грамматические навыки – это фундамент, без которого может легко разрушиться всё хрупкое здание коммуникативности. Механический подход к усвоению грамматических конструкций (иначе говоря, простое заучивание парадигмы спряжения того или иного глагола, изолированного от контекста или речевой ситуации) имеет ряд недостатков.

Речь идёт об отсутствии навыка самостоятельно, грамматически правильно конструировать предложения. Разумное использование грамматических игр, как надёжное средство формирования грамматических навыков, реализуется учителями английского языка на практике.

Аудитивные игры

Цели:

- научить учащихся понимать смысл однократного высказывания;
- научить учащихся выделять главное в потоке информации;
- развить слуховую память учащихся.

Данные игры представляют собой восприятие информации учениками на слух.

Речевые игры

Цели:

- научить учащихся умению выражать мысли в их логической последовательности;
- научить учащихся практически и творчески применять полученные речевые навыки;
- обучить учащихся речевой реакции в процессе коммуникации.

Самый эффективный вид игры для обучения говорению – это сюжетно-ролевая игра. С её помощью дети учатся переносить действия из одних условий в другие, начинают одушевлять неживую природу, создавать элементы нового. Так развиваются воображение, мышление, творческие способности, речь, воспитываются нравственно-волевые качества личности, интерес к личности другого.

Более того, благодаря сюжетно-ролевым играм происходит формирование **критического мышления**, т. е.

- комплекс когнитивных навыков и умений;
- открытое мышление, развивающееся путем наложения новой информации на жизненный опыт;
- отражение оценочных и рефлексивных свойств мышления;

У учащихся формируются когнитивные навыки и умения:

- выражать свои мысли (устно и письменно) ясно, уверенно, корректно;
- аргументировать свою точку зрения и учитывать мнения других;
- оценить информацию;
- уметь задавать вопросы;
- объяснить изученное и применить в практике.

Игры для обучения чтению

Цели:

- быстро и точно устанавливать букво-звуковые соответствия;
- правильно озвучивать графический образ слова и соотносить его со значением, т.е. понимать читаемое;
- читать по синтагмам, объединяя слова в определенные смысловые группы;
- читать в естественном темпе текст, построенный на знакомом языковом материале, с пропущенными словами или рисунками вместо слов;
- выразительно, с правильным ударением и интонацией читать вслух (парное обращенное чтение).

Игровые упражнения для обучения чтению – это упражнения на закрепление буквенно-звуковых связей, творческие задания на установление соответствия предлагаемых рисунков содержанию текста.

Взаимосвязь чтения с другими коммуникативными навыками и умениями может осуществляться посредством изобразительной деятельности.

Таким образом, игры способствуют выполнению важных методических задач:

- созданию психологической готовности детей к речевому общению;
- обеспечению естественной необходимости многократного повторения ими языкового материала;
- тренировке учащихся в выборе нужного речевого варианта, что является подготовкой к ситуативной спонтанности речи вообще.

ТРУДНОСТИ В ОВЛАДЕНИИ ПИСЬМЕННОЙ РЕЧЬЮ НА АНГЛИЙСКОМ ЯЗЫКЕ НА СТАРШЕМ ЭТАПЕ ОБУЧЕНИЯ

Кондратьева Марианна Яновна

Решение задач, связанных с обучением письменной речи на иностранном языке, сопряжено с целым рядом трудностей.

Первый комплекс трудностей связан с длительностью процесса овладения письменной формой речи, а также недостаточным учебным временем, выделяемым программой для решения этой задачи. Другой комплекс трудностей связан с неисследованностью письменной речи как дидактико-методического объекта в связи с невостребованностью его как целевого вида речевой деятельности, недооценкой потенциальных возможностей письменной речи для языкового и интеллектуального развития личности, а также сложностью междисциплинарной интерпретации письменной речи как многокомпонентного, многоуровневого процесса.

Трудности обучения письменной речи объясняются психологической сложностью. А.А.Миролюбов говорит о том факте, что "включая в себя все те нервно-мозговые связи, которые необходимы для владения устной речью, письменная речь требует образования целого ряда дополнительных ассоциаций". Известно, что к слухо-речедвигательным связям, с помощью которых осуществляется устная речь, при письме добавляются зрительно-слуховые и речедвигательно-моторно-графические связи.

Сложность письменной речи связана не только с необходимостью включения большого числа анализаторов. Её намного усложняют условия, в которых она обычно протекает.

Как свидетельствуют многочисленные исследования Л.С.Выготского, С.А.Рубинштейна, одним из обстоятельств, усложняющих акт письменной речи, является отсутствие ситуации, общей для пишущего и того, кому адресована речь. Таким образом, то, что при устном общении может быть недосказано или вовсе опущено, при письменном общении должно найти свое полное выражение.

Письменная речь должна быть максимально ясной и развернутой, так как она выполняет коммуникативную функцию. Отметим то обстоятельство, что восприятие читающего не зеркально отражает ход развития мысли у пишущего и поэтому без понимания мыслительного процесса последнего и без знания и видения путей превращения замысла в письменное высказывание невозможно научиться процессу письменного выражения мыслей.

Задачей пишущего является создание текста, но при этом пишущий должен помнить об адресате. Я.М.Колкер указывает на тот факт, что в сознании пишущего, в соответствии с мотивом и целью, а также на основании предыдущего опыта и учета особенностей восприятия адресата, возникает общий замысел всего текста, включающий в себя его свернутое общее содержание. Этот общий замысел текста обычно выражен в приблизительной формулировке темы письменного высказывания.

В момент возникновения замысла текста у пишущего нет еще полного представления о его частях и деталях. Это одна из трудностей, с которой сталкивается учащийся. Выбор частей, их пересмотр и отбрасывание происходит в зависимости от степени их соответствия или несоответствия оси общего замысла, которая должна проходить через всю деятельность

пишущего, то есть так или иначе общий замысел, пусть формально и не выраженный, должен присутствовать в каждой части текста. При этом все части текста должны относиться друг к другу как отдельное к отдельному в рамках общего.

Каждая коммуникативно-значимая часть, прямо или косвенно характеризующая объект высказывания, должна находиться относительно другой части текста на более высоком или более низком уровне членения, в зависимости от степени важности по отношению к объекту высказывания того признака, который в ней уточняется или расчленяется. Текст имеет важное значение, когда мы говорим о письменной речи. Текст – готовый продукт, и только рассмотрение его с точки зрения хода развития мыслей и путей и способов их детализации превращает анализ этого текста в упражнение, которое обучает пишущего предугадывать предполагаемые вопросы читателя и давать на них правильные ответы, что и является процессом порождения письменного высказывания.

В настоящее время, процесс обучения иностранному языку, построенный на коммуникативной основе, расширяет возможности предмета. Поэтому, при обучении письменной речи важно придерживаться коммуникативного подхода, так как обучение ориентировано на личность и строится таким образом, что непосредственная деятельность учеников, их опыт, мировоззрение, учебные и внеучебные интересы и склонности учитываются при организации общения на уроке, является ли это письменным или устным общением.

Стоит обратить внимание на то, что если устная речь предусматривает постоянную стимуляцию и контроль со стороны слушателя, то эти моменты при письменной речи отсутствуют. Кроме того, если при устном общении говорящий может воспользоваться рядом вспомогательных средств (жесты, мимика, интонация), то при использовании письменной речи таких возможностей не существует.

Механизм письменного составления текста не является простым. В работах психологов отмечается, что процесс составления письменного текста включает в себя такие элементы как отбор слов, необходимых для данного конкретного текста, распределения предметных признаков в группе предложений, выделение предиката как стержневой части в смысловой организации предложения, организацию связи между предложениями. При этом основной действующей силой при составлении письменного текста считается его упреждение, то есть представление о том, что будет написано, еще до того момента написания.

Отбор слов зависит от запаса слов пишущего, от его словарного фонда и умения правильно его использовать при решении конкретной коммуникативной задачи. Следовательно, недостаточно иметь богатый словарный запас слов, необходимо научиться пользоваться ими. Слова, которые могли употребляться в данном тексте, сами по себе им известны, но в текст не включаются. Расширению словарного запаса при составлении текстов может служить, например, коммуникативно-составленный подбор в плане будущего текста с подбором ключевых слов, выражений, типовых фраз к каждому пункту текста.

На всех ступенях обучения, особенно на старших, следует обращать внимание учащихся на стилистические особенности и употребление тех или иных слов. Правильность письменной речи складывается из правильности каждого предложения в отдельности и правильности их в контексте.

Говоря о письменной речи, необходимо подчеркнуть тот факт, что письменное перефразирование мыслей является одной из трудностей в овладении письменной речью. Это сложный умственный процесс. Он формирует у учащихся умения и навыки быстро и успешно производить необходимый отбор слов как с целью более точной передачи мысли, так и с целью избегания повторения одних и тех же слов и предложений.

По ряду психологических соображений, связанных со школьными условиями обучения иностранному языку, центр тяжести в работе над письменной речью должен быть перенесен на работу учащихся дома.

Характер заданий по письменной речи следует ставить в зависимость от таких факторов как степень языковой подготовленности учащихся и возраст учащихся.

О первом факторе говорилось ранее. Что же касается возраста учащихся – это одна из проблем, которая не всегда учитывается в должной степени учителями, не соблюдается необходимая градация в работах по письменной речи.

Из педагогической психологии известно, что от возраста учащихся во многом зависит степень их творческой самостоятельности в письменной речи. Младшие школьники значительно охотнее повторяют услышанное, чем продуцируют сами. Совершенно иначе поступают учащиеся старших классов. Они стараются полностью осмыслить содержание услышанного или прочитанного, а затем уже передать его по возможности своими словами.

Назначение периода юности в жизни каждого человека состоит в том, чтобы расширить горизонты познания реального мира, других людей, себя самого, выработать ко всему своё отношение, найти своё место в обществе и определить жизненные задачи.

Центральным психологическим процессом юношеского возраста является развитие самосознания.

Старший школьник миновал время подростковых кризисов и конфликтов. В этом возрасте отмечается улучшение коммуникативности и общего эмоционального самочувствия индивида.

Учащиеся характеризуются возросшей общительностью. Общение в этом возрасте играет важную роль. В нём старшеклассники ищут и находят доказательства своей социальной и человеческой ценности.

Письменная речь является одним из современных средств общения и учащиеся старших классов проявляют интерес к нему.

Ранее говорилось о том, что старшеклассники передают или стараются передать услышанное и прочитанное своими словами. При этом изменяется сам характер упреждения текста во внутренней речи: текст составляется не по отдельным предложениям, а по смысловым частям. Внешние связи между отдельными предложениями и частями текста заменяют связями внутреннего, логического порядка.

Необходимо учесть, что при оформлении связанного письменного высказывания, когда внимание учащихся смещается с графико-орфографической формы слов на смысловую сторону высказывания, трудности возрастают. Ученики должны в этом случае составить план-программу высказывания, отобрать из долговременной памяти языковые средства, свойственные письменной речи, произвести необходимые замены, комбинирование, составить целое из частей и затем развернуть высказывание в структуру целого текста во внешней речи.

Выполнение всех этих операций вызывает у учеников большие трудности. Преодоление этих трудностей требует ориентированного в этих целях обучения.

ПЕДАГОГИЧЕСКИЕ НАУКИ

ДИСТАНЦИОННОЕ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ ПЕДАГОГИЧЕСКИХ РАБОТНИКОВ В МБОУ «ЕЛАНЦЫНСКАЯ СОШ»

Васильева Ирина Михайловна

Подготовка высококвалифицированных педагогических кадров - одна из ключевых проблем системы образования России. Развитие информационных технологий и их повсеместное внедрение в систему образования требует от педагога мобильности, гибкости, способности идти в ногу со временем, в том числе при совершенствовании своей профессиональной компетентности[1].

Если до сих пор повышение квалификации педагогических кадров происходило один раз в пять лет, то в настоящее время педагогам значительно чаще требуется повышение уровня профессиональной компетентности. При учете скорости осуществления реформ в системе образования становится очевидным, что существующая сегодня система дополнительного педагогического образования не в состоянии удовлетворить как требования государства к системе повышения квалификации работников образования, так и актуальные информационные потребности педагогов. Переход традиционной системы повышения квалификации педагогических кадров на личностно-ориентированное основание, создание педагогических условий активизации профессионального развития и саморазвития педагога, повышение эффективного управления процессом повышения квалификации на основе новых информационных технологий является одним из путей преодоления кризиса в системе повышения квалификации [2]. С одной стороны, педагог должен быть готов к постоянному совершенствованию и повышению своей квалификации, а с другой стороны – обществом должны быть созданы условия, при которых педагог может реализовать свою потребность в постоянном обучении и развитии. Для того чтобы в течение всего периода профессиональной деятельности соответствовать постоянно растущим квалификационным требованиям, педагогическим работникам необходимо обучаться на курсах повышения квалификации при региональных институтах повышения квалификации [3].

Актуальным становится повышение квалификации педагогических работников через дистанционное обучение, которое в определенной мере обеспечивает решение проблем дополнительного педагогического образования, а именно: (а) доступ неограниченного количества педагогических работников к повышению квалификации; (б) быструю и качественную подготовку педагогических работников к управлению инновационными процессами, вызванными модернизацией российского образования; (в) обучению без отрыва от основного места работы и возможности непосредственного использования полученных знаний в профессиональной деятельности; (г) реализации принципов индивидуализации и дифференциации обучения; (д) снижению затрат на обучение за счет сокращения расходов, связанных с командированием и замещением педагогических работников.

Необходимость учета возможностей и потребностей слушателей системы повышения квалификации, появление новых профессиональных задач, стоящих перед работниками образования, и отсутствие разработанной и отлаженной системы повышения квалификации, учитывающей нарастающую информатизацию социальной и образовательной жизни общества, актуализируют данное направление деятельности.

Проблема внедрения информационных технологий и использования дистанционного обучения в образовательном процессе затронута в работе известных ученых: Е.С. Полат, А.Е. Петрова [4]. Из работы ученых следует, что создание эффективных систем дистанционного обучения создает условия социальной доступности к качественному образованию для значительной части населения, содействует решению проблемы образования для людей, которые по различным причинам не могут воспользоваться услугами очного обучения. Дистанционное обучение является специфической формой получения образования, которая, может наряду с другими формами эффективно использоваться в системе непрерывного профессионального образования.

На данном этапе модернизации образования в Ольхонском районе назрела необходимость в организации информационной поддержки дистанционного обучения педагогических работников, главной целью которой является предоставление слушателям полной и достоверной информации о возможных путях повышения квалификации посредством дистанционного образования. Результатом внедрения сетевой модели будет создание таких условий, при которых педагогические работники смогут выбирать курс, разрабатывать индивидуальные образовательные маршруты, что будет способствовать качеству работы не только самого учителя, но и школы в целом.

В районе вопросами повышения квалификации педагогических работников занимается районный ресурсный центр (РРЦ)

РРЦ организует и курирует дистанционное повышение квалификации, проводит мониторинг потребностей в области повышения квалификации, организует консультативную поддержку педагогов образовательных учреждений области в сфере использования ИКТ в образовательном процессе.

Всего в районе на 1 сентября 2011/2012 учебного года работает 154 педагогических работника в 9 общеобразовательных учреждениях.

Районный ресурсный центр, расположенный в здании МБОУ «Еланцынская СОШ» совместно с ИИПКРО и ОРЦ ежегодно проводит семинары для начинающих пользователей персональным компьютером (ПК). За последние четыре года прошли обучение в рамках семинара «Начинающий пользователь ПК» более 90% учителей-предметников района, из них 70% - педагогические работники МБОУ «Еланцынская СОШ». Успешно освоившие материалы семинара «Пользователь ПК» стали слушателями дистанционных курсов, предоставляемых в рамках повышения квалификации учреждениями повышения квалификации Иркутской области ОГАОУ ДПО ИРО и ИПКРО (Рис. 2.). РРЦ занимается предоставлением образовательных ресурсов слушателям, согласно их запросам.

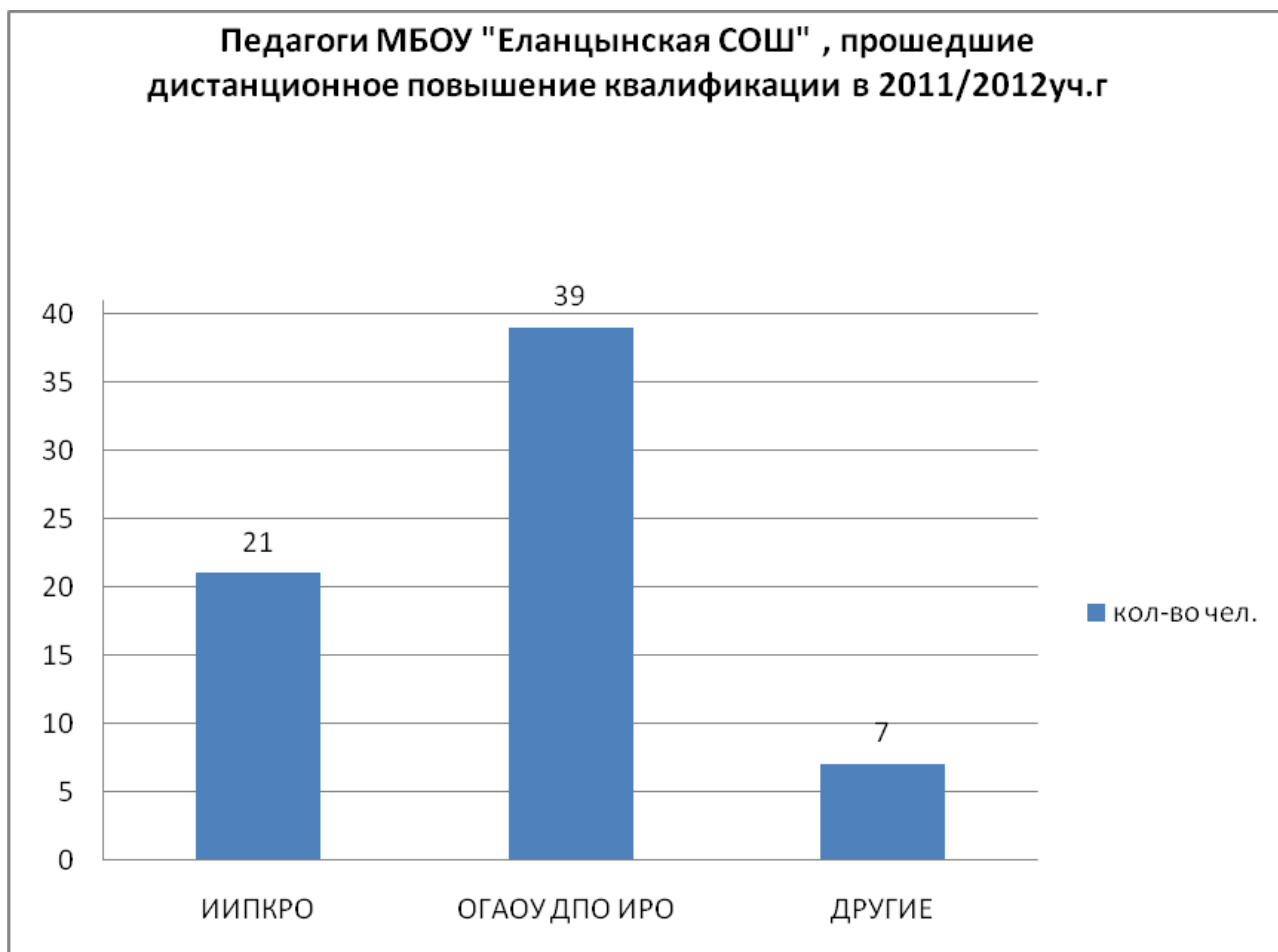


Рис.2. Дистанционное повышение квалификации педагогов МБОУ «Еланцынская СОШ» в 2011/2012 уч.году.

Районным ресурсным центром разрабатывается сетевая модель повышения квалификации в форме дистанционного обучения (Рис.3), внедрение которой позволит помочь педагогическим работникам с выбором курса для организации повышения квалификации в форме дистанционного обучения. Районным ресурсным центром проводится следующая работа: прием заявок от педагогических работников; диагностирование (составление диагностических карт ОУ, которые выявляют проблемные вопросы, темы, для каждого педагога ОУ); индивидуальное собеседование с целью создания индивидуальных маршрутов; изучение адресных заявок, содержащих индивидуальные запросы по актуальным проблемам повышения квалификации; систематизация и анализ выявленных проблем – запросов педагогов; определение поля взаимодействия с институтом повышения квалификации (ИИПКРО) или институтом развития образования (ОГАОУ ДПО ИРО), а также другими организациями; консультация и техническое сопровождение во время обучения педагога; выдача удостоверений.



Рис.3. Сетевая модель повышения квалификации педагогических работников в МБОУ «Еланцынская СОШ» в форме дистанционного обучения.

В рамках выполнения требования по аттестации педагогических работников приоритетными являются курсы по учебным предметам и ИКТ.

За последние два года наиболее востребованы курсы ИИПКРО (предметные) и ИРО (информационно-коммуникационные технологии), так как имеют определяющее значение для прохождения аттестации на квалификационную категорию.

Техническая подготовка учебного процесса в МБОУ «Еланцынская СОШ» представлена следующими условиями для развития дистанционного обучения: в образовательном учреждении есть все необходимые технические условия (компьютерный парк составляют 40 компьютеров, подключенных к локальной сети); постоянное подключение к сети Интернет (скорость 512 Кбит/с., наземное подключение, ООО «Иркутскэнергосвязь», безлимит, 2500 руб/мес.).

Мотивация, как уже отмечалось авторами работы, является одним из важнейших факторов обеспечения качества дистанционного повышения образования. Около 80% педагогических работников имеют навыки работы в области информационных технологий, из них 9% имеют собственные сайты.

Входные и итоговые анкетирования, проводимые РРЦ среди 58 слушателей при определении мотивации к повышению квалификации в форме дистанционного обучения показали следующие позиции: удовлетворение собственного интереса (57%), потребность в росте самооценки (10%), желание карьерного роста (28%), жесткие требования руководства образовательного учреждения («заставили») (5%).

Возрастная структура педагогических кадров на сегодняшний день является одной из важных характеристик состояния школьной образовательной системы. По результатам опросов, РРЦ наблюдается явное увеличение процента педагогов предпенсионного и пенсионного возраста. Среди педагогических работников МБОУ «Еланцынская СОШ» ситуация выглядит следующим образом: моложе 25 лет – 3%; от 25-35 лет – 11%; 40 лет и старше – 86%, из них 18% – пенсионеры.

Это еще раз подтверждает, что в повышении самооценки более заинтересованы преподаватели старшего возраста, это можно объяснить желанием «не отставать от более молодых коллег и обучаемых» и является хорошим стимулом к повышению квалификации.

Из общей численности педагогических работников (61 чел.) имеют стаж работы: менее 2 лет – 3%; от 5 до 10 лет – 5%; от 10 до 20 лет – 29%; 20 лет и более – 63%.

За последних три года прошли дистанционные предметные курсы (ИИПКРО) более 50% преподавателей МБОУ «Еланцынская СОШ» разных специальностей, в основном те, кто находится в возрастной категории 40 лет и старше и стаж работы которых составил 20 лет и более.

Как показывают данные анкетирования, дистанционные курсы повышения квалификации вызывают живой интерес в школах района, в том числе и в МБОУ «Еланцынская СОШ». Однако следует отметить, лишь 16% коллектива педагогических работников МБОУ «Еланцынская СОШ» ежегодно повышают квалификацию в форме дистанционного обучения

Педагоги, использующие для повышения квалификации дистанционное обучение, в большей степени управляют своим самообразованием, что соответственно говорит об их профессиональном росте, все они имеют высшую категорию. Они также отмечают главные достоинства дистанционной формы самообразования: возможность пройти в удобное время; возможность выбора темы по интересующим и наиболее актуальным вопросам.

Среди недостатков, выделяют: многие дистанционные курсы проводятся на платной основе; документы, подтверждающие факт прохождения дистанционного обучения, чаще всего не имеют юридической силы, т. е. их не учитывают при проведении очередной аттестации за исключением курсов организованных ИИПКРО и ИРО.

Выводы:

Анализ и обобщение полученных результатов в процессе внедрения сетевой модели повышения квалификации в форме дистанционного обучения на базе РРЦ показывают эффективность предложенных нововведений, заключающихся в: широком охвате педагогических работников Ольхонского района; индивидуальном подходе в выборе дистанционного курса повышения квалификации; экономии времени на повышение квалификации педагогических работников за счет отсутствия поездок в учебные заведения и использовании сетевой модели дистанционного обучения; экономической эффективности (уменьшение материальных затрат на базовые, транспортные расходы).

Заключение

Дистанционное обучение педагогических работников с целью непрерывного образования может и должна занять свое место в системе повышения квалификации, поскольку при грамотной организации дистанционного обучения возможно обеспечение качественного образования, соответствующего требованиям современного общества сегодня и ближайшей

перспективы.

Говоря о дистанционном обучении как форме повышения квалификации, следует говорить о создании единого информационно-образовательного пространства, куда следует включить всевозможные электронные источники информации (включая сетевые): виртуальные библиотеки, базы данных, консультационные службы, электронные учебные пособия и т.д. Когда речь идет о дистанционном обучении следует понимать наличие в системе преподавателя, курса и слушателя.

Отсюда следует, что главным при организации дистанционной формы обучения является создание электронных курсов, разработка дидактических основ дистанционного обучения, подготовка преподавателей-координаторов, разработка индивидуального маршрута для слушателя.

Список литературы

1. Концепция долгосрочного социально-экономического развития РФ до 2020 г. [Электронный ресурс] — 194с. <http://www.ifap.ru/ofdocs/rus/rus006.pdf> (29 марта 2009г.).
2. Молчанов С. Г. Профессиональная компетентность в системе повышения квалификации / С. Г. Молчанов // Интеграция методической работы и системы повышения квалификации кадров. — Челябинск, 2003. — 54 с.
3. Хачиров С. В. Дистанционное повышение квалификации педагогов на базе сети районных ресурсных центров: дис. ... канд. пед. наук: 13.00.08 / С. В. Хачиров. — СПб., 2005. — 168с.
4. Полат Е. С., Петров А. Е. Концепция дистанционного обучения на базе компьютерных телекоммуникаций в России. URL: <http://www.distant.ioso.ru/library/publication/concept.htm>

АНАЛИЗ ДЕЯТЕЛЬНОСТИ ОЛЬХОНСКОГО РАЙОННОГО РЕСУРСНОГО ЦЕНТРА С ПОЗИЦИЙ СИСТЕМНОГО ПОДХОДА

Васильева Ирина Михайловна

Зарождение системного подхода относится к XIX в. и связано с трудами К.Маркса (общественные или социальные системы), Ч.Дарвина (биологические системы), Д. И. Менделеева (периодическая система химических элементов). В разработку идей системного подхода значительный вклад внесли также Ч. Барнард, Г. Саймон, Р.Аккофф, С.Оптнер, Д. Клиланд, У.Кинги и др.

Хотя существует множество конкретных моделей системного подхода в управлении, общим для них является стремление рассматривать *организацию* не как простую сумму каких-то отдельных компонентов, а как систему, т. е. как *совокупность связанных и взаимодействующих частей, как целое, обладающее собственными качествами, отличными от качеств составляющих его частей*.

Таким образом, *системный подход* в управлении образованием предполагает изучение объектов и явлений в их целостности, в их взаимосвязи с другими явлениями.

Практика показывает, что применение системного подхода в управлении образованием вызвано необходимостью:

- рассмотрения управляемых объектов образования и явлений, связанных с образовательной деятельностью и нуждающихся в управленческом вмешательстве, в единстве частного и целого;
- создания надежной основы для классификации управляемых объектов и явлений для выбора эффективных методов и технологий управленческой деятельности;
- объединения ситуационно разрозненных фактов образовательной практики в единую систему для принятия правильного управленческого решения.

По мнению известного исследователя общей теории систем Дж. ван Гига, системный подход является порождением целостной оптимизации задачи, решаемой в условиях развития сложных систем.

Системный подход в управлении ориентирует управленца на *оптимизацию* (улучшение) объекта и явления в целом, с учетом всего многообразия проблем их функционирования и развития, а не на оптимизацию их отдельных частей, и представляет собой альтернативу методу частичного улучшения объектов и явлений.

В отечественной научной литературе термином «оптимизация» обозначается процесс, позволяющий получать наивысший возможный для конкретных условий результат.

С позиций системного подхода любая организация (в нашем случае — это районный ресурсный центр) является открытой системой. Она взаимодействует с внешней средой, обменивается с ней энергией, информацией, материалами. Эффективность организации определяется не только ее системными качествами, но и условиями среды. Эти условия постоянно меняются, поэтому, чтобы оставаться эффективной, организация должна развиваться и приобретать

новые качества. Изменения ее частей не должны осуществляться изолированно, они потребуют согласованных изменений и других частей.

Ранее подходы к управлению были обращены «внутрь управляемой организации». Системный же подход прежде всего рассматривает управляемую организацию как часть другой, внешней по отношению к этой организации, большей системы. Например, общеобразовательная школа как системный объект понимается как часть социума микрорайона, в котором она находится, или как часть муниципальной системы образования города, или как часть региональной, федеральной систем и т.д. Только затем с позиций системного подхода выделяется то, что определяет «поведение» системного объекта, его особенности.

При системном подходе к управлению в центре внимания оказывается процесс принятия решений. Решения принимаются на всех уровнях организации, в том числе и на уровне рядовых работников (они решают, работать или не работать в этой организации, какие усилия прикладывать, как строить отношения с коллегами и администрацией и т.д.).

Практика показывает, что при реализации системного подхода в управлении необходимо осуществить:

- анализ цели управленческой деятельности как системного объекта;
- определение назначения создаваемой системы управления по отношению к другим системам (внешним или более высокого порядка) и ее места в системе профессиональной деятельности, осуществляемой в изучаемом объекте;
- выявление ее структуры и протекание в ней процессов;
- определение критериев эффективности создаваемой системы;
- раскрытие методов и прогнозов развития (в этом плане системный подход должен отождествляться с проектированием).

Построить систему управления — значит сформулировать цель управленческой деятельности, выделить совокупность элементов системы и установить характер связей между ними, которые позволяют реализовать управленческие действия в достижении поставленной цели.[1]

Создание эффективной системы образования требует внедрения новых подходов, использующих современные информационные технологии в области обработки и применения информации в образовательном процессе, и современного менеджмента в сфере управления качеством образования. **Повышение эффективности системы образования, возрастание качества и доступности образовательных услуг базируются на внедрении использования современных информационных и коммуникационных технологий в образовательные учреждения района.**

Администрация Ольхонского района и управление народного образования приняли решение пять лет назад, что в районе необходимо такое учреждение, в котором будет сконцентрирована работа с педагогическими работниками и учащимися, где будет осуществляться дополнительная подготовка выпускников для поступления в ВУЗы и реализовываться программы дополнительного образования с использованием компьютерной техники.

Для этих целей было выделено помещение на базе МБОУ «Еланцынская СОШ», в котором было установлено пятнадцать компьютеров и 6 марта 2007 года открыт Районный ресурсный центр. Районный ресурсный центр информационной и научно-методической поддержки образования (далее - РРЦ) - структурное подразделение Управления народного образования и создан на основании распоряжения (постановления) мэра (главы администрации) от 8.12.2006 г. № 452 (Положение на сайте РРЦ- <http://trc-olhon.ucoz.ru>)

Основными целями Ресурсного центра являются:

- развитие информационного образовательного пространства и распространения информационной культуры в образовательной среде Ольхонского муниципального района;
- научно-методическая и информационная поддержка федеральных и областных целевых программ в сфере информатизации образования;
- просветительская работа с педагогическими и руководящими кадрами муниципальных образовательных учреждений.

С 2007 года РРЦ работает над решением следующих задач: методическое сопровождение целевых федеральных, региональных и муниципальных программ образования и воспитания; оказание методической поддержки образовательным учреждениям в освоении и введении в действие Федеральных государственных образовательных стандартов (ФГОС) второго поколения; оказание помощи в развитии творческого потенциала педагогических работников образовательных учреждений; удовлетворение информационных, учебно-методических, образовательных потребностей педагогических работников образовательных учреждений; создание условий для организации и осуществления повышения квалификации педагогических и руководящих работников образовательных учреждений; оказание методической и консультационной поддержки всем участникам образовательного процесса. создание и обслуживание базы образовательных электронных ресурсов (ОЭР); сопровождение районных, областных мероприятий в сфере образования: видеоконференций, проблемных семинаров, научно-практических и методических конференций, конкурсов, олимпиад, проектов и других форм методической работы; консультационная и аналитическая работа: предоставление информации по запросам области и муниципальных органов управления образованием, ОРЦ, образовательных учреждений, КПМО; сопровождение государственной итоговой аттестации (ЕГЭ, ГИА). организация районных конкурсов, а также сопровождение областных, федеральных для воспитанников детских садов, обучающихся школ и педагогических работников.

В 2012-2013 учебном году РРЦ продолжалась работа по реализации национальной образовательной инициативы «Наша новая школа». Особое значение уделялось развитию учительского потенциала - повышению уровня компетентности педагогических кадров.

Работа по повышению квалификации педагогов проводилась в сотрудничестве с ИПКРО и ИРО. Формы работы РРЦ по повышению квалификации педагогов разнообразны: семинары, вебинары, мастер-классы, конференции, видеоконференции, курсы повышения квалификации, в том числе дистанционные.

Только в текущем году на базе Ресурсного центра были организованы следующие курсы повышения квалификации: «Внедрение пакета свободного программного обеспечения», ИИПКРО с 12.09.11 – 24.09.11 (72 ч), 16 учителей, МБОУ «Еланцынская СОШ»; «Внедрение пакета свободного программного обеспечения», ИИПКРО в дистанционном режиме (72 ч), 2 педагога дошкольного образования «Ромашка»; «Основы компьютерной грамотности», «Твой курс», ИИПКРО (18 ч) в дистанционном режиме, 6 учителей, МБОУ «Еланцынская СОШ», 3 педагога дошкольного образования «Ромашка», 1 социальный педагог, 4 учащихся МБОУ «ЕСОШ»; «Создание ЭОР для дистанционного обучения», ИРО, с 15-16 мая 2012г., 14 учителей, МБОУ «Еланцынская СОШ»; «Сайтостроение», ИПКРО, с 25.06.2012, 72 часа, 21 учитель, МБОУ «Еланцынская СОШ», МКОУ «Алагуевская ООШ».

В течение 2011 – 2012 г. выехали на курсы повышения квалификации учителя и педагоги дошкольного образования по следующим темам: «Формирование системы духовно-нравственных ценностей средствами музыки» - ИИПКРО – 72 ч (1 учит ЕСОШ); «Основы религиозных культур и светской этики», ИИПКРО, 72 ч. (8 уч.- ЕСОШ, БСОШ, ЧСОШ, ХСОШ, АООШ, ОСОШ, КСОШ); «Основы трехмерного моделирования» в дистанционном режиме,

ИИПКРО 72 ч. (3 уч.- ЕСОШ – 2, ХСОШ -1); «Новые педагогические технологии на основе использования современных форм аттестации выпускников. Технологические принципы использования обучающего дидактического тестирования с применением ИКТ» ИРО 72 ч. (4 уч.- ОСОШ, ХСОШ, АООШ, ЕСОШ); и т.д.

А также, на базе Ресурсного центра в течение последнего года прошли уроки, заседания методических объединений, руководителей ОУ, классные часы и внеклассные мероприятия: «Введение. Техника безопасности. Информация. Ввод информации» учитель – Романова Н.Г. 8 класс; «Word. Ввод текста» учитель – Романова Н.Г. 8 класс; Открытый классный час, 8 класс, учитель Хамадаева Э.В.; МХК «Проверочная работа», учитель – Долгонова Г.Г. 11 класс; Читательская конференция 10-е классы МБОУ ЕСОШ в рамках недели русского языка и литературы к юбилею Распутина В.Г., учитель Дамбуева Г.А.; Открытый урок 10 класс в рамках недели английского языка, МБОУ ЕСОШ, учитель Афонькина Л.П.; Заседание творческой группы «Применение инновационных технологий в образовательном процессе в средней школе», руководитель Баннова Т.С.; «Космическое путешествие» среди детских садов; Внеклассное мероприятие среди 5-х классов «Своя игра. Древняя Греция», учитель Долгонова Г.Г., Открытые уроки в рамках недели информатики и математики в 5,7,9,10 классах, учитель Васильева И.М.

На адрес школ ежемесячно рассылаются темы проводимых вебинаров. За последний год педагогические работники МБОУ «Еланцынская СОШ» на базе РРЦ в дистанционном режиме, приняли участие в шести вебинарах.

Для обучающихся района сотрудниками ресурсного центра проводятся различные конкурсы и олимпиады, причем не только школьного и районного масштаба, но и областные, международные и т.д. В десятый раз в этом учебном году проходил фестиваль «Информационные технологии». Отрадно, что уже третий год он стал межтерриториальным.

В РРЦ школьники могут пройти различные курсы, причем не только по овладению компьютером и программами, но и ежегодные дистанционные курсы по русскому языку и математике, которые ведут учителя базовой школы района для старшеклассников Алагуевской, Чернорудской, Тонтинской, Бугульдейской и Хужирской школ. Также ученики могут пройти дистанционное обучение, организованное РРЦ совместно с Новосибирской дистанционно математической и физической школой, Иркутским государственным университетом (ЗФМШ), и ИРО («Телешкола»). Все это направлено, в первую очередь, на высокую подготовку старшеклассников к экзаменам.

Что касается экзаменов, то в РРЦ проходят уже несколько лет традиционные репетиционные ЕГЭ и ГИА. Чтобы экзамен не был неожиданностью для ребенка, в центре проводятся различные тестирования и пробные экзамены. Все организационные вопросы ложатся на плечи работников центра, в том числе сбор информации, выверка данных для базы данных по выпускникам.

В центре также проводятся различные конкурсы для педагогов, тематические мероприятия к различным праздникам. Кроме этого, ресурсный центр сам принимает активное участие в конкурсах.

Теплые отношения поддерживает коллектив с коллегами других районов области. В 2009 году РРЦ радушно принял на Ольхонской земле руководителей и методистов территориальных ресурсных центров. Согласно плана ОРЦ, был проведен семинар "Деятельность территориальных ресурсных центров по созданию единого информационного пространства Иркутской области".

С сентября 2009 года пополняется муниципальный банк данных «Одаренные дети».

В текущем учебном году необходимо приобрести программу «1С: Управление образованием 3.0».

Использование многофункциональной системы «Управление образованием 3.0» позволит комплексно решить задачу автоматизации управления системой образования на районном уровне. Это означает автоматизацию потоков информации из всех МОУ и ДОУ, с организацией полного информационного обмена между ними и муниципальными органами управления на основании единых информационных стандартов.

Таким образом, внедрение специального программного обеспечения на платформе «1С», т.е. информационной комплексной системы в управление образованием является важной частью предметного содержания образовательного менеджмента. Системный же подход к использованию информационного пространства повышает эффективность управленческих решений.

Список литературы

1. Воробьева С.В. Основы управления образовательными системами. – М.: Академия, 2008.
2. Давыденко Т.М., Шамова Т.И., Шибанова Г.Н. Управление образовательными системами. – М.: Академия, 2007.
3. Сайт Ольхонского районного ресурсного центра <http://trc-olhon.ucoz.ru>

ЛИЧНОСТЬ УЧИТЕЛЯ В ФОРМИРОВАНИИ ЦЕННОСТНО-СМЫСЛОВЫХ КОМПЕТЕНЦИЙ МЛАДШИХ ШКОЛЬНИКОВ

Питина Людмила Сергеевна

Основной тенденцией, способствующей эффективности профессионально-личностного развития учителей начальных классов, выступает тенденция взаимообусловленности формирования профессиональной деятельности и личности педагога. Принципиальная смена ценностно-смысловых ориентиров современного образования ведет за собой полноценное развитие личности и включение ее в социально ценную деятельность, предполагающую различные возможности самообразования. В этой связи и педагогический процесс должен основываться на реализации ряда основополагающих принципов ценности и уникальности личности, ориентации на зону ближайшего развития, субъектности и эмоционально-ценностной ориентации учебного процесса [1].

При лидирующей значимости ориентиров на личностный результат образования содержательная и процессуальная специфика формирования личностных ценностно-смысловых компетенций школьников предъявляет особые требования к личности учителя как основного транслятора ценностей и медиатора эмоционально-волевого развития школьников.

Ценностный тип образования влияет на структуру и содержание ценностных ориентации педагогов, задает то или иное понимание смыслов педагогической деятельности. Поскольку ценности воспринимаются детьми через образы конкретных личностей, наделенных качествами, адекватными данным ценностям, доверительные отношения младшего школьника с авторитетным воспитателем позволяют считать последнего объектом для подражания и ориентироваться на него в своем поведении, что указывает на многозначное влияние личности воспитателя на эмоционально-ценностное образование детей.

Только аутентичный учитель может помочь проявиться аутентичности (естественности) ребенка, когда поведение и речь правдивы и непосредственно выражают внутренние чувства. У такого учителя ребенок обнажается психологически и духовно и позволяет увидеть себя вместо того, чтобы прятаться (А. Маслоу). Если ребенок ценен для педагога только как ученик, то ценностно-смысловая сфера личности (внутренний мир ребенка) перестает быть психологическим объектом нравственного, духовного воспитания, так как подобное воспитание не может осуществляться посредством воздействий лишь вербального характера. Чаще всего это ненаправленное влияние, когда учитель становится для школьника значимым существом, источником смыслов и ценностей.

Ориентируясь на эмоциональное отношение учителя к тому, что изучается на уроке, ребенок усваивает не только информацию, но и ее оценку взрослым. Оценки учителя, его отношение к ученику в значительной степени определяют самооощущение ребенка, его представление о своих возможностях и способностях. Вера в учителя, доверие к нему создают необходимые психологические условия для превращения детей в воспитуемых. Практика воспитания убеждает: ничто так быстро не вызывает недоверия к педагогу, как расхождение между тем, что он говорит, в чем убеждает, и тем, каков его собственный внутренний мир.

Мировидение, мироотношение учащихся в значительной мере определяются и теми

ценностными ориентациями, которые стали достоянием их внутреннего мира и на «вращивание» которых, несомненно, повлияли ценности взрослых. Хорошо, если эти ценности направлены на ребенка в его главной роли – Человека, на процесс, а не только на результат и форме ЗУНов. Не случайно понятие педагогического профессионализма рассматривается в контексте того, в какой мере педагог может развить субъектный потенциал учащихся, обеспечить условия его личностного роста (В.В. Давыдов, Ю.Б. Орлов, В.А. Сластенин, И.С. Якиманская и др.). А это возможно лишь при условии, что личностное развитие ребенка входит в систему ценностей педагогической культуры учителя.

Высокий новаторский уровень педагога, реализующего опыт ценностно-смысловых отношений, определяется готовностью к обновлению мировосприятия, обоснованной смене оценок, их детализации и развитию. Мы согласны с исследователями, утверждающими, что ориентация на процесс – это философско-педагогическое кредо учителя, ценностно направленного на гуманизацию и гуманитаризацию образовательной среды и человеческих (субъект-субъектных) отношений участников образовательного процесса, что предполагает создание условий для решения наиважнейших проблем в жизнедеятельности любого образовательного учреждения – проблем общения и отношений (В.Г. Воронцова, С.И. Маслов и др.).

Ценностно - смысловые компетенции - это компетенции, связанные с ценностными ориентирами ученика, его способностью видеть и понимать окружающий мир, ориентироваться в нем, осознавать свою роль и предназначение, уметь выбирать целевые и смысловые установки для своих действий и поступков, принимать решения. Данные компетенции обеспечивают механизм самоопределения ученика в ситуациях учебной и иной деятельности. От них зависит индивидуальная образовательная траектория ученика и программа его жизнедеятельности в целом.

Ориентируясь на условия, необходимые для формирования личностных компетенций младших школьников, сама педагогическая деятельность приобретает специфические черты, среди которых важно предвидение и проектирование влияния любого педагогического воздействия на развитие эмоционально-волевой и ценностно-смысловой сферы младших школьников.

Особого внимания заслуживает то обстоятельство, что основным фактором формирования системы ценностей у отдельно взятой личности является внутренний фактор, т.е. личностные структуры сознания, формирующие личностный опыт создания собственной картины мира. При этом они инициируются внешними воздействиями как специально организованная деятельность учителя, «работающего» с учеником через преобразование содержания учебного материала.

Учитель в значительной мере транслирует информацию посредством чувств, эмоций. Поэтому восприятие учителя определяется, в большей степени, личностью педагога, его способностью эмоционально стимулировать воздействие сообщаемой информации, произвольно воздействовать на непроизвольные эмоциональные реакции учащихся.

Идея использования эмоциональных состояний учителя для решения педагогических задач является одной из форм реализации личностного подхода в образовании (Ш.А. Амонашвили, И.П. Волков, О.С. Газман, Е.Н. Ильин, В.А. Караковский, А.В. Мудрик, Л.И. Новикова, В.Ф. Шаталов и др.). Достижение эмоционального контакта является наиболее важным звеном в деятельности учителя как условие эмоциональной активации учебного процесса.

Взаимосвязь участников образовательного процесса осуществляется через передачу на вербальном и невербальном уровне знаковой информации. Это может способствовать оптимальному становлению ценностной сферы детей на примере учителя, т.к. имидж учителя

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММНЫЕ СРЕДСТВА В СИСТЕМЕ ВЫСШЕГО ОБРАЗОВАНИЯ

Баранова Ольга Александровна
Жужжалов Валерий Евгеньевич

Компьютерное обучение является качественно новым видом обучения, все стороны которого далеко не раскрыты. Важным достоинством компьютерной технологии обучения является существенное расширение информационной основы обучения – компьютерной информационной среды, включающей базы данных, знаний, моделей, оценочных процедур, диагностических и контрольных тестов и др. В отличие от многих давно известных технических средств обучения, включение компьютера в учебный процесс, сопровождаемое практически неограниченными возможностями информационной поддержки, изменяет традиционную роль средств обучения и существенно влияет на структуру учебного процесса.

Современные информационные технологии в настоящее время позволяют по-новому подойти к проблеме индивидуализации обучения. Использование программного компьютерного продукта, основанного на принципах искусственного интеллекта, оказалось перспективным направлением в осуществлении личностно-ориентированного обучения.

Несмотря на существующий опыт, проблемы создания и использования программных средств учебного назначения не перестают быть актуальными. Их принято делить на:

- психологические, отражающие вопросы взаимодействия учащегося и компьютера, особенности их диалога, индивидуальные особенности и оценку когнитивного стиля учащегося и др.;
- методические, рассматривающие место компьютера в учебном процессе, целесообразность использования программных средств на различных его этапах, выбор программных средств, режим применения, отбор содержания, способы оценки эффективности и др.;
- педагогические, анализирующие педагогические условия компьютеризации обучения, вопросы реализации личностно-ориентированного обучения и способы повышения эффективности образовательного процесса;
- физиолого-гигиенические, определяющие комфортные и безопасные условия работы учащихся с компьютерной техникой, влияющие на протекание учебной деятельности;
- технологические, затрагивающие вопросы структуры программных компьютерных средств, их типологию, способы построения моделей процесса обучения и обучаемого, разработка стратегий обучения и др.;
- технические, включающие выбор средств для создания программных компьютерных систем, разработку алгоритмов и программ, их тестирование, отладку и т.п.;
- организационно-экономические, связанные с организацией разработки, оценки эффективности, тиражирования, продвижения на рынке образовательных технологий и т.п.

К настоящему времени создано множество компьютерных программных средств, применяемых в образовательном процессе: контролирующие, тренажерные, моделирующие и демонстрационные программы; автоматизированные обучающие системы; гиперсреды, мультимедиа и программы виртуальной реальности; электронные учебники; интеллектуальные

и экспертные обучающие системы. Компьютерные технологии все активнее используются для поддержки и расширения возможностей традиционных методов обучения. Важным ресурсом для реализации личностно-ориентированного метода обучения выступает, в частности, электронный (компьютерный) учебник.

Электронный учебник – это не обычный печатный учебник в виде его электронной копии. Нелинейная гипертекстовая структура содержания электронного учебника коренным образом отличает его от обычного – печатного, где всегда доминирует линейный принцип. Электронный учебник с элементами искусственного интеллекта является эффективным обучающим средством, имеющим множество достоинств. Обучаемому он может предлагать маршрут в гипертексте учебной предметной области, задавать вопросы, получать ответы, использовать различные обучающие воздействия, контролировать качество обучения, выполнять некоторые иные функции, например, справочные. Другими словами – человек и компьютер могут вести продуктивный диалог в режиме обучения.

Несмотря на достоинства и дидактические возможности современных программных средств учебного назначения, в существующей практике многие достижения, реализующие концептуально новые решения в области разработки компьютерных информационных технологий, используются недостаточно. Методика применения компьютерных технологий основана, как правило, на эмпирических подходах, а компьютерные программные средства, составленные с учетом требований индивидуализации обучения в технологической подготовке студентов практически отсутствуют.

В настоящее время наметился системный подход к решению проблемы индивидуализации обучения. Задачу учета индивидуальных особенностей обучающихся, их познавательных интересов, способностей и возможностей способна решить реализация личностно-ориентированной модели обучения.

Для осуществления личностно-ориентированного обучения с позиций его индивидуализации необходимы следующие основные предпосылки:

- уровневая дифференциация содержания образования (разные варианты программ, учебников, дидактических материалов, позволяющих на едином базовом содержании знаний варьировать и индивидуализировать процесс обучения);
- новые формы и методы проведения групповых и индивидуальных занятий;
- постоянное внимание к анализу и оценке способов учебной работы ученика, побуждающих его к осознанию не только результатов, но и процесса своей работы;
- специальная подготовка преподавателя к систематическому осуществлению такой работы;
- развитие рефлексии на свои собственные способы работы и их совершенствование.

Существенный вклад в решение задач уровневой дифференциации могут внести внедряемые сейчас новые образовательные стандарты и информационные технологии обучения. При этом, с одной стороны, минимальный базовый уровень учебной подготовки, предусмотренный стандартами, обеспечивает доступность образования каждому, с другой стороны, создаются условия для более полного раскрытия личных возможностей и способностей всех учащихся.

Новая волна интереса к проблеме индивидуализации была вызвана внедрением новых информационных технологий (НИТ) в обучение. Многие исследователи НИТ считают, что в современных условиях индивидуализацию обучения можно эффективно развивать с помощью компьютерных технологий.

В соответствии с изменившейся концепцией образования современные подходы к обучению в высшем образовании должны быть направлены на достижение целей, ориентированных

непосредственно на личность каждого обучаемого. Индивидуализированная технологическая подготовка, обеспечивающая достижение поставленных целей, призвана решить следующие задачи:

- развить способности и интересы учащихся;
- сформировать технологические знания, практические умения и навыки безопасной работы;
- повысить самостоятельность в процессе технологической подготовки;
- закрепить в практической деятельности знания, полученные при изучении основ наук;
- выявить скрытые (потенциальные) возможности отстающих и создать условия для развития их способностей;
- выявить наиболее способных студентов, которые в будущем составят резерв одаренных аспирантов и ассистентов вузов технического профиля.

Методы обучения в широком плане, относящиеся ко всем учебным дисциплинам, составляют предмет исследования дидактики. В задачи методик преподавания входит разработка применения уже исследованных дидактикой общих методов обучения с учетом особенностей содержания и научных методов исследования той или иной науки. Однако использование уникальных возможностей компьютерной технологии не может быть самоцелью, так как замена других средств обучения компьютерной программой не всегда может быть педагогически оправданной. Мотивировать компьютерное обучение можно только реальными потребностями действующей или вновь создаваемой методики обучения.

Список литературы

1. Жужжалов В.Е. Основы интеграции парадигм программирования в курсе информатики. — М.: «Образование и Информатика», Институт содержания и методов обучения РАО, 2006г. —127с.
2. Баранова О.А. Способы обработки информации и программирования в высшей школе. — LAP LAMBERT Academic Publishing, ISBN: 978-3-8484-4480-9, Россия, 2012г. —69с.

ПСИХОЛОГИЧЕСКИЕ НАУКИ

ИЗУЧЕНИЯ СКЛОННОСТЕЙ К ОПРЕДЕЛЕННОМУ СТИЛЮ РУКОВОДСТВА У СТУДЕНТОВ С РАЗНЫМ УРОВНЕМ ЛИДЕРСКОГО ПОТЕНЦИАЛА

Соколова Ольга Николаевна

Проблема поиска детерминант успешного лидерства традиционно относится к основным прикладным проблемам социальной и организационной психологии. В связи с глобальными изменениями социального уклада общества в конце XX начале XXI века резко возрос спрос на лидеров - личностей, способных объединять других людей для достижения поставленных задач - в рамках организаций, неформальных объединений и политических движений. Можно констатировать, что в настоящий момент существует несомненная потребность конкретных организаций и общества в целом в специалистах, обладающих способностями к реализации лидерского поведения.

Лидерство и руководство — два основополагающих понятия, с которыми связано эффективное управление организациями.

В 1937 году К. Левин выделил общую психологическую составляющую деятельности всех лидеров (руководителей), назвав ее лидерством. Основная часть работ по проблеме лидерства появилась в 70-е годы. Также этой проблемой в психологии занимались такие психологи как: Р.М. Стогдилл, Ф. Фидлер; Г. Кунц и С.О. Доннел; Т. Митчел и Р. Хаус, П. Херси и К. Бланшард.

Давая свои объяснения стилям лидерства, современные исследователи неизменно связывали их с деятельностью группы (Г. Гибш, М. Форверг), подразумевая под ними способы, с помощью которых осуществляется функция руководства в группе (Г. Гибш).

Вопросами типологии стилей руководства занимались психологи в разное время, начиная с 30-х годов и до настоящего времени. Так К. Левин, Р. Липитт (1939 г.), Г. Гибш и М. Форверг (1939 г.), К. Бирт, Ф. Прильвиту (1959 г.), Р. и А. Тауш (1963 г.), Х. Хрншель (1964 г.) придерживались точки зрения, согласно которой стали обозначаться такие стили руководства, как авторитарный, демократический, анархический.

Психология управления как прикладная наука пытается найти связь между стилем руководства и продуктивностью процесса управления. Ведь только компетентный, авторитетный руководитель, имеющий отработанные навыки профессионального общения с членами группы создает в ней благоприятный психологический климат и сплачивает сотрудников для выполнения организационных целей. [1]

Что касается лидерского потенциала, то до сих пор остается проблематичным определение необходимых и достаточных индивидуальных характеристик личности руководителя, которые бы способствовали успешности деятельности руководителя-лидера.[3] Руководитель как активный субъект управленческой деятельности, включенный одновременно в различные сферы деятельности (профессиональную, организационную, управленческую, общественную), должен обладать субъектными и личностными качествами, объединенными общей направленностью (на дело, на результат, на людей, на организацию, на общество). Эти качества, присущие ему как субъекту, должны рассматриваться как новые интегративные качества. Таким качеством, по нашему предположению, может быть лидерский потенциал.

Важно отметить тот факт, что независимо от того, какие именно стили управления руководителей наиболее эффективно можно будет использовать при работе в конкретном коллективе, руководитель должен постоянно улучшать свои лидерские навыки и работать не только над совершенствованием руководящего процесса и взаимоотношений с сотрудниками, но и над собой, повышая и развивая свой лидерский потенциал. Таким образом, лидерский потенциал может быть основой способности личности исходить из устойчивых внутренних критериев и ориентиров в своей жизнедеятельности, сохранять стабильность управленческой деятельности и смысловых ориентаций на фоне давлений и изменяющихся внешних условий.

Наша работа в данном исследовании строилась на предположении о том, что существует связь между показателями стиля руководства и уровнем лидерского потенциала, то есть чем выше лидерский потенциал, тем будет выше склонность к демократическому стилю руководства, а также чем ниже лидерский потенциал, тем будет выше склонность к авторитарному стилю руководства.

Мы исследовали студентов 4-5х курсов факультета управления и предпринимательства Белгородского государственного университета. Всего участвовало 30 человек.

В процессе изучения склонностей к определенному стилю руководства нами была использована следующая методика «Склонность к определенному стилю руководства» Е. П. Ильина. Для изучения определения лидерского потенциала мы использовали методику «Определение уровня лидерского потенциала». Полученные данные исследования обработаны с помощью ранговой корреляции Спирмена.

В результате эмпирического исследования нами была установлена взаимосвязь между стилем руководства и лидерским потенциалом.

Мы выяснили, что существует положительная сильная связь $0,85(p=0,01)$ между показателями демократического стиля и уровнем лидерского потенциала. Таким образом, чем выше лидерский потенциал, тем будет выше склонность к демократическому стилю руководства. Успешный лидер не стремится к власти, он ею пользуется; также он не стремится брать на себя ответственность, но берет, потому что лучше всех способен ею распорядиться.

Также имеем сильную отрицательную связь $0,85(p=0,01)$ между показателями авторитарного стиля руководства и уровнем лидерского потенциала. То есть чем ниже лидерский потенциал, тем будет выше склонность к авторитарному стилю руководства. Можно сказать, что авторитарный руководитель, нацеленный на власть, имеет ее реально меньше, чем при любой другой цели, поскольку не обладает лидерским потенциалом.

В результате нашего эмпирического исследования нами была установлена взаимосвязь между стилем руководства и лидерским потенциалом.

Список литературы

1. Журавлев, А.А. Стиль в современной психологии управления / А.А. Журавлев – М.: Издание "Наука", 1996.- 340 с.
2. Зинченко, В.П., Моргунов, Е.Б. Человек развивающийся. Очерки российской психологии. / В.П. Зинченко, Е.Б. Моргунов. – М.: Наука, 1994 - 400 с.
3. Кудряшова, Л.Д. Каким быть руководителю./ Л.Д. Кудряшова. – Л.:1986 г.,- с.164.

Содержание

Технические науки	1
КРАТКАЯ ХАРАКТЕРИСТИКА СЕТЕВЫХ КОМАНД ОПЕРАЦИОННОЙ СИСТЕМЫ WINDOWS	2
СОСТАВ КОМАНД WINDOWS	28
РОЛЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И РЕСУРСОВ В МАРКЕТИНГЕ	34
СЕТЕВЫЕ УТИЛИТЫ ОС WINDOWS	36
КОНВЕЙЕРИЗАЦИЯ КОМАНД WINDOWS	49
ПРАКТИЧЕСКОЕ ИСПОЛЬЗОВАНИЕ СИСТЕМЫ КОМАНД WINDOWS	52
КЛАССЫ .NET ДЛЯ РАБОТЫ С ПРОЦЕССАМИ	74
СОЗДАНИЕ ПРОЦЕССОВ И УПРАВЛЕНИЕ ИМИ	85
ЗАПУСК КОМАНД ОС ИЗ C#-ПРОГРАММЫ	89
СПРАВОЧНЫЕ ДАННЫЕ НЕКОТОРЫХ КОМАНД	104
Экономические науки	110
ЭКОНОМИЧЕСКАЯ СОЦИОЛОГИЯ КАК НАУКА	111
СОВРЕМЕННАЯ ЗАПАДНАЯ ЭКОНОМИЧЕСКАЯ СОЦИОЛОГИЯ	114
ПРИМЕНЕНИЕ ТЕОРИИ ХАОСА К АНАЛИЗУ РЫНКОВ	117
УЧЕТ КАК СОВРЕМЕННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА	119
АНТИКРИЗИСНОЕ УПРАВЛЕНИЕ ФИНАНСАМИ ПРИ УГРОЗЕ БАНКРОТСТВА	122
Филологические науки	125
РОЛЬ ИГРЫ НА УРОКАХ АНГЛИЙСКОГО ЯЗЫКА	126
ТРУДНОСТИ В ОВЛАДЕНИИ ПИСЬМЕННОЙ РЕЧЬЮ НА АНГЛИЙСКОМ ЯЗЫКЕ НА СТАРШЕМ ЭТАПЕ ОБУЧЕНИЯ	130
Педагогические науки	133
ДИСТАНЦИОННОЕ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ ПЕДАГОГИЧЕСКИХ РАБОТНИКОВ В МБОУ «ЕЛАНЦЫНСКАЯ СОШ»	134
АНАЛИЗ ДЕЯТЕЛЬНОСТИ ОЛЬХОНСКОГО РАЙОННОГО РЕСУРСНОГО ЦЕНТРА С ПОЗИЦИЙ СИСТЕМНОГО ПОДХОДА	140
ЛИЧНОСТЬ УЧИТЕЛЯ В ФОРМИРОВАНИИ ЦЕННОСТНО-СМЫСЛОВЫХ КОМПЕТЕНЦИЙ МЛАДШИХ ШКОЛЬНИКОВ	145
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММНЫЕ СРЕДСТВА В СИСТЕМЕ ВЫСШЕГО ОБРАЗОВАНИЯ	148
Психологические науки	151
ИЗУЧЕНИЯ СКЛОННОСТЕЙ К ОПРЕДЕЛЕННОМУ СТИЛЮ РУКОВОДСТВА У СТУДЕНТОВ С РАЗНЫМ УРОВНЕМ ЛИДЕРСКОГО ПОТЕНЦИАЛА	152

