

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования
«Гомельский государственный университет имени Франциска Скорины»
Факультет физики и информационных технологий
Кафедра общей физики

Отчёт по лабораторной работе №4
«Обработка текста в ОС Windows»

Выполнил:
студент группы КИ-22: Аксёнов И. С.

Проверил: Грищенко.В.В

Гомель 2025

Цель работы: Изучить основные принципы работы с текстом и текстовыми файлами в Windows.

Краткие сведения из теории

Стандартный ввод, вывод и вывод ошибок

Многие программы операционных систем выводят свои результаты на экран терминала. Однако правильное понимание процесса вывода информации предполагает знание о стандартных файлах ОС. Существует как минимум 3 стандартных файла:

- стандартный файл ввода (stdio);
- стандартный файл вывода (stdout);
- стандартный файл ошибок (stderr).

Эти три стандартных файла доступны любому пользовательскому процессу с момента его возникновения. При работе пользовательского процесса можно организовать вывод в файл stdout (standard output), а сообщения о состоянии – в специальный файл стандартный вывод ошибок stderr (standard error). По умолчанию оба файла, стандартный вывод и стандартный вывод ошибок, связаны с экраном и не сохраняются на диске. Кроме того, многие программы принимают ввод из специального файла с названием стандартный ввод stdin (standard input), который по умолчанию связан с клавиатурой.

Механизм перенаправления ввода/вывода позволяет изменять направление вывода и ввода. Обычно вывод осуществляется на экран, а ввод – с клавиатуры, но механизм перенаправления ввода/вывода позволяет изменить этот порядок вещей.

Механизм перенаправления ввода/вывода позволяет явно указать, куда должен осуществляться стандартный вывод. Чтобы перенаправить стандартный вывод в другой файл вместо экрана, нужно добавить в команду оператор перенаправления `>` и имя файла. Где это может пригодиться? Иногда полезно сохранить вывод команды в файл. Например, можно сообщить командной оболочке, что она должна направить вывод команды `ls` в файл `ls-output.txt` вместо экрана:

```
$ ls -l /usr/bin > ls-output.txt
```

Здесь мы создали длинный список содержимого файла `/usr/bin` и отправили результаты в файл `ls-output.txt`. Простой оператор перенаправления, без предшествующей ему команды, очистит существующий файл или создаст новый, пустой файл. Для добавления вывода в конец существующего файла используем оператор перенаправления `>>`:

```
$ ls -l /usr/bin >> ls-output.txt
```

Перенаправление стандартного вывода ошибок осуществляется не так просто, как стандартного вывода. Чтобы перенаправить стандартный вывод ошибок, нужно указать его дескриптор файла. Программа может производить вывод в любой из нескольких нумерованных файловых потоков. Первые три из них мы упомянули как стандартный ввод, вывод и вывод ошибок. Командная оболочка ссылается на них как на файловые дескрипторы 0, 1 и 2 соответственно. Командная оболочка поддерживает синтаксис перенаправления файлов с использованием номеров файловых дескрипторов. Так как стандартному выводу ошибок соответствует файловый дескриптор 2, мы можем перенаправить его, как показано ниже:

```
$ ls -l /bin/usr 2> ls-error.txt
```

Номер файлового дескриптора 2 помещается непосредственно перед оператором перенаправления, чтобы перенаправить стандартный вывод ошибок в файл `ls-error.txt`. Иногда необходимо сохранить весь вывод команды в один файл. Для этого перенаправьте сразу два потока, стандартный вывод и стандартный вывод ошибок. Сделать это можно двумя способами. Первый — традиционный — работает в старых версиях командной оболочки:

```
$ ls -l /bin/usr > ls-output.txt 2>&1
```

Здесь выполняется два перенаправления. Сначала — перенаправление стандартного вывода в файл `ls-output.txt`, а затем, с использованием нотации `2>&1`, — перенаправление файлового дескриптора 2 (стандартный вывод ошибок) в файловый дескриптор 1 (стандартный вывод).

Современные версии `bash` поддерживают второй, более простой метод выполнения перенаправления этого вида:

```
$ ls -l /bin/usr &> ls-output.txt
```

В данном примере используется единственный оператор `&>`, перенаправляющий стандартный вывод и стандартный вывод ошибок в файл `ls-ou.tput.txt`

Обработка текстовой информации в Windows

В PowerShell, как и в командной строке есть способ перенаправить поток вывода и ошибок в файл. Как и в командной строке в PowerShell существуют операторы перенаправления, аналогичные используемым в Linux. Кроме того существуют и другие потоки (таблица 1)

Таблица 1 – Стандартные потоки в PowerShell

Поток	Варианты использования	Описание
0		Поток ввода
1	<code>>filename »filename</code>	Поток информационных сообщений
2	<code>2>filename 12>>filename 12>&1</code>	Поток сообщений с ошибками
3	<code>3> filename 3>>filename 13>&1</code>	Поток предупреждений.
4	<code>4>filename 14>>filename 14>&1</code>	Поток подробного вывода.
5	<code>5> filename 15>>filename 15>&1</code>	Поток диагностики.
*	<code>*>filename *>>fitename</code>	Все потоки.

Следующие инструменты предназначены для работы с текстом в Windows:

- New-Item – позволяет создать текстовый файл.
- Get-Content – позволяет получать данные из текстовых файлов.
- Set-Content, Add-Content – заменяет или добавляют объекты в файл.
- Out-File – записывает данные передаваемые по конвейеру в файл.
- Tee-Object – сохраняет отображаемые объекты и передаёт их дальше по контейнеру.
- Export-CSV, Import-Csv – предназначены для работы с csv-файлами.
- Export-Clixml, Import-Clixml – предназначен для работы с xml-файлами

Ход работы

```
PS C:\Users\Administrator> Get-ChildItem -Path $HOME | Out-File output.txt
PS C:\Users\Administrator> Get-Content output.txt

Directory: C:\Users\Administrator
```

Mode	LastWriteTime		Length	Name
----	-----	-----	-----	----
d-r----	21.05.2025	23:06		Contacts
d-----	22.05.2025	16:29		current
d-r----	21.05.2025	23:06		Desktop
d-----	22.05.2025	16:26		dir1
d-r----	21.05.2025	23:06		Documents
d-r----	21.05.2025	23:06		Downloads
d-r----	21.05.2025	23:06		Favorites
d-r----	21.05.2025	23:06		Links
d-r----	21.05.2025	23:06		Music
d-----	22.05.2025	16:28		new
d-----	22.05.2025	16:28		old
d-r----	21.05.2025	23:06		Pictures
d-----	22.05.2025	16:26		qqq
d-r----	21.05.2025	23:06		Saved Games
d-r----	21.05.2025	23:06		Searches
d-----	22.05.2025	17:35		sorted
d-----	22.05.2025	17:38		unsorted
d-r----	21.05.2025	23:06		Videos
-a----	22.05.2025	16:14	0	dir2
-a----	22.05.2025	16:14	0	dir3
-a----	22.05.2025	16:35	306	directories.zip
-a----	22.05.2025	18:24	0	output.txt
-a----	22.05.2025	17:46	866	task.completed.zip

Рисунок 1 - Запись содержимого домашнего каталога в output.txt и вывод результата на экран

```
PS C:\Users\Administrator> (Get-Content output.txt) -replace '[,.-]', '+' | Out-File result.txt
PS C:\Users\Administrator> Get-Content result.txt

Directory: C:\Users\Administrator
```

Mode	LastWriteTime		Length	Name
++++	+++++	+++++	+++++	++++
d+r+++	21+05+2025	23:06		Contacts
d+++++	22+05+2025	16:29		current
d+r+++	21+05+2025	23:06		Desktop
d+++++	22+05+2025	16:26		dir1
d+r+++	21+05+2025	23:06		Documents
d+r+++	21+05+2025	23:06		Downloads
d+r+++	21+05+2025	23:06		Favorites
d+r+++	21+05+2025	23:06		Links
d+r+++	21+05+2025	23:06		Music
d+++++	22+05+2025	16:28		new
d+++++	22+05+2025	16:28		old
d+r+++	21+05+2025	23:06		Pictures
d+++++	22+05+2025	16:26		qqq
d+r+++	21+05+2025	23:06		Saved Games
d+r+++	21+05+2025	23:06		Searches
d+++++	22+05+2025	17:35		sorted
d+++++	22+05+2025	17:38		unsorted
d+r+++	21+05+2025	23:06		Videos
+a++++	22+05+2025	16:14	0	dir2
+a++++	22+05+2025	16:14	0	dir3
+a++++	22+05+2025	16:35	306	directories+zip
+a++++	22+05+2025	18:24	0	output+txt
+a++++	22+05+2025	17:46	866	task+completed+zip

Рисунок 2 - Замена знаков препинания (,-) на + и вывод результата замены на экран

```

PS C:\Users\Administrator> Get-PSDrive | Out-File hdd.txt
PS C:\Users\Administrator> Get-Process | Out-File pl.txt
PS C:\Users\Administrator> Get-CimInstance -ClassName Win32_Processor | Out-File cpu.txt

PS C:\Users\Administrator> Get-Process -Id 320 | Out-File process_320.txt
PS C:\Users\Administrator> Get-NetTCPConnection | Out-File tcp.txt
PS C:\Users\Administrator> Get-NetUDPEndpoint | Out-File udp.txt
PS C:\Users\Administrator> Get-NetRoute | Out-File net.txt

PS C:\Users\Administrator> netstat -s | Out-File nstat.txt

PS C:\Users\Administrator> Get-NetTCPConnection | Where-Object { $_.State -eq 'Listen' } | Out-File open_ports.txt
PS C:\Users\Administrator> |

```

Рисунок 3 - Сбор системной информации

```

PS C:\Users\Administrator> Get-Process | Where-Object { $_.CPU -gt 50 } | Sort-Object CPU -Descending | Out-File high_utilization_processes.txt

```

Рисунок 4 - Управление процессами

```

Get-ChildItem -Path "C:\Program Files" -Recurse | Where-Object { $_.Length -gt 100MB -and $_.LastWriteTime -gt (Get-Date).AddDays(-7) } | Out-File large_recent_files.txt

```

Рисунок 5 - Поиск файлов по размеру и дате изменения

```

PS C:\Users\Administrator> Get-ChildItem -Path "C:\Users\Administrator\Documents" -Recurse | Select-String "error" | Out-File error_logs.txt
PS C:\Users\Administrator> |

```

Рисунок 6 - Поиск файлов с содержимым "error"

```

PS C:\Users\Administrator> Get-ChildItem -Path "C:\Users\Administrator\Documents" -Recurse -ErrorAction SilentlyContinue | Where-Object { $_.Length -gt 1GB } | Sort-Object
PS C:\Users\Administrator> S

```

Рисунок 7 - Анализ использования дискового пространства

```

PS C:\Users\Administrator> Get-PSDrive | Select-Object Name, User, Free | Sort-Object Used -Descending | Tee-Object -FilePath disk_report.txt

```

Name	User	Free
Alias		
C		42162192384
Cert		
D		0
Env		
Function		
HKCU		
HKLM		
Variable		
WSMan		

Рисунок 8 - Создание отчета о дисковом пространстве

Контрольные вопросы

1. Для чего используются текстовые файлы в ОС?

Текстовые файлы применяются для хранения и обмена данными в удобном для человека формате. Они используются в конфигурационных файлах, логах, документации, программном коде, а также для межпроцессного взаимодействия.

2. Что такое стандартные файлы ОС?

Стандартные файлы операционной системы — это потоки ввода, вывода и ошибок (stdin, stdout, stderr). Они используются программами для приема данных, вывода информации и обработки ошибок.

3. Как осуществляется перенаправление работы стандартных файлов?

Перенаправление осуществляется с помощью операторов командной строки:

- `>` — перенаправление вывода в файл (перезапись);
- `>>` — добавление вывода в файл;
- `<` — ввод данных из файла;
- `2>` — перенаправление потока ошибок.

4. Какие утилиты имеются в Windows для работы с текстом?

В Windows есть множество инструментов, включая:

- Notepad и Notepad++ — текстовые редакторы;
- `type` и `more` — просмотр содержимого файлов в командной строке;
- `find` и `findstr` — поиск текста в файлах;
- `echo` — вывод текста в командной строке.

5. Как произвести сохранение вывода любой команды в файл?

Можно использовать оператор `>` для записи вывода команды в файл. Например:

```
shell
```

```
dir > список_файлов.txt
```

Эта команда сохранит список файлов текущего каталога в `список_файлов.txt`.

6. Как сделать замену символов в текстовом файле?

В Windows это можно сделать с помощью утилиты PowerShell. Например, для замены "старый" на "новый" в файл.txt используйте команду:

```
powershell
```

```
(Get-Content файл.txt) -replace "старый", "новый" | Set-Content файл.txt
```

Это заменит все вхождения "старый" в файле и сохранит изменения.