

Отчёт по лабораторной работе №8

Операционные системы

Пономарева Татьяна Александровна

Содержание

1	Цель работы	5
2	Теоретическое введение	6
2.1	Перенаправление ввода-вывода	6
3	Выполнение лабораторной работы	7
4	Ответы на контрольные вопросы	10
5	Выводы	13
	Список литературы	14

Список иллюстраций

3.1	Вход в систему	7
3.2	Записывание названий файлов в file.txt	7
3.3	Записывание названий файлов в conf.txt	7
3.4	Определение файлов с 'с' в домашнем каталоге	7
3.5	Вывод файлов из /etc, начинающихся с 'h', постранично	8
3.6	Запуск процесса в фоновом режиме и удаление ~/logfile	8
3.7	Запуск gedit в фоновом режиме и определение его идентификатора	8
3.8	Завершение процесса gedit	8
3.9	Информация о df и du, их выполнение	9
3.10	Вывод всех директорий в домашнем каталоге	9

Список таблиц

1 Цель работы

Ознакомление с инструментами поиска файлов и фильтрации текстовых данных. Приобретение практических навыков: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

2 Теоретическое введение

2.1 Перенаправление ввода-вывода

В системе по умолчанию открыто три специальных потока: - `stdin` — стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0; - `stdout` — стандартный поток вывода (по умолчанию: консоль), файловый дескриптор 1; - `stderr` — стандартный поток вывод сообщений об ошибках (по умолчанию: консоль), файловый дескриптор 2. Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода `stdout`.

3 Выполнение лабораторной работы

Вхожу в систему при помощи команды `whoami` - выводит имя моей учетной записи (`taponomareva`) (рис. 3.1).

```
[taponomareva@taponomareva ~]$ whoami
taponomareva
```

Рис. 3.1: Вход в систему

Записываю названия файлов из каталога `/etc` и домашнего каталога в `file.txt` при помощи команд `ls /etc > file.txt` и `ls ~ >> file.txt` (рис. 3.2).

```
[taponomareva@taponomareva ~]$ ls /etc > file.txt
[taponomareva@taponomareva ~]$ ls ~ >> file.txt
```

Рис. 3.2: Записывание названий файлов в `file.txt`

Вывожу файлы с расширением `.conf` и записываю их в `conf.txt`, используя `grep` `".conf$"` `file.txt > conf.txt` (рис. 3.3).

```
[taponomareva@taponomareva ~]$ grep "\.conf$" file.txt > conf.txt
```

Рис. 3.3: Записывание названий файлов в `conf.txt`

Определяю файлы, начинающиеся с `'с'` в домашнем каталоге, совершая команды `ls ~ | grep "^с"` и `find ~ -maxdepth 1 -type f -name "с*"` (рис. 3.4).

```
[taponomareva@taponomareva ~]$ ls ~ | grep "^с"
conf.txt
[taponomareva@taponomareva ~]$ find ~ -maxdepth 1 -type f -name "с*"
/home/taponomareva/conf.txt
```

Рис. 3.4: Определение файлов с `'с'` в домашнем каталоге

Вывожу имена файлов из /etc, начинающихся с 'h', постранично: `ls /etc | grep "^h" | less` (рис. 3.5).

```
[taonomareva@taonomareva ~]$ ls /etc | grep "^h" | less
host.conf
hostname
hosts
(END)
```

Рис. 3.5: Вывод файлов из /etc, начинающихся с 'h', постранично

Запускаю процесс в фоновом режиме, записывающий файлы, начинающиеся с 'log', в ~/logfile: `ls ~ | grep "^log" > ~/logfile &`; затем удаляю файл ~/logfile: `rm ~/logfile` (рис. 3.6).

```
[taonomareva@taonomareva ~]$ ls ~ | grep "^log" > ~/logfile &
[1] 3832
[taonomareva@taonomareva ~]$ rm ~/logfile
[1]+  Done                  ls --color=auto ~ | grep --color=auto "^log" > ~/logfile
```

Рис. 3.6: Запуск процесса в фоновом режиме и удаление ~/logfile

Запускаю редактор gedit в фоновом режиме при помощи команды `gedit &` и определяю его идентификатора процесса: `ps aux | grep gedit` или `pgrep gedit` (рис. 3.7).

```
[taonomareva@taonomareva ~]$ gedit &
[1] 2358
[taonomareva@taonomareva ~]$ ps aux | grep gedit
taonom+ 2358 10.4 0.6 773528 58356 pts/0 S1 01:04 0:00 gedit
taonom+ 2384 0.0 0.0 238340 2200 pts/0 S+ 01:04 0:00 grep --color=auto gedit
[taonomareva@taonomareva ~]$ pgrep gedit
2358
```

Рис. 3.7: Запуск gedit в фоновом режиме и определение его идентификатора

Завершаю процесса gedit, используя `kill` (подставьте номер процесса) или `pkill gedit` (рис. 3.8).

```
[taonomareva@taonomareva ~]$ kill 2358
[taonomareva@taonomareva ~]$ pkill gedit
```

Рис. 3.8: Завершение процесса gedit

Получаю информации о df и du, затем их выполняю: `man df` - справка по команде df, информация о свободном и используемом пространстве файловых систем;

man du - справка по команде du, оценка занимаемого дискового пространства файлами и директориями; df -h - показывает свободное место на дисках (читаемый формат); du -sh ~ - показывает размер домашней директории в удобном формате(рис. 3.9).

```
[taponomareva@taponomareva ~]$ man du
[taponomareva@taponomareva ~]$ man df
[taponomareva@taponomareva ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        79G   14G   65G   18% /
devtmpfs         4.0M    0  4.0M    0% /dev
tmpfs            4.1G   348K   4.1G    1% /dev/shm
tmpfs            1.7G   1.2M   1.7G    1% /run
tmpfs            1.0M    0  1.0M    0% /run/credentials/systemd-udev-load-credentials.service
tmpfs            1.0M    0  1.0M    0% /run/credentials/systemd-journald.service
tmpfs            1.0M    0  1.0M    0% /run/credentials/systemd-network-generator.service
tmpfs            1.0M    0  1.0M    0% /run/credentials/systemd-sysctl.service
tmpfs            1.0M    0  1.0M    0% /run/credentials/systemd-tmpfiles-setup-dev-early.service
tmpfs            1.0M    0  1.0M    0% /run/credentials/systemd-tmpfiles-setup-dev.service
/dev/sda3        79G   14G   65G   18% /home
tmpfs            4.1G   4.0K   4.1G    1% /tmp
tmpfs            1.0M    0  1.0M    0% /run/credentials/systemd-vconsole-setup.service
/dev/sda2       974M   321M   586M   36% /boot
tmpfs            1.0M    0  1.0M    0% /run/credentials/systemd-tmpfiles-setup.service
tmpfs            1.0M    0  1.0M    0% /run/credentials/systemd-resolved.service
tmpfs           824M   88K   824M    1% /run/user/1000
[taponomareva@taponomareva ~]$ du -sh ~
1.2G    /home/taponomareva
```

Рис. 3.9: Информация о df и du, их выполнение

Вывожу все директории в домашнем каталоге: find ~ -type d (рис. 3.10).

```
[taponomareva@taponomareva ~]$ find ~ -type d
/home/taponomareva/.password-store/.git/objects/95
/home/taponomareva/.password-store/.git/logs
/home/taponomareva/.password-store/.git/logs/refs
/home/taponomareva/.password-store/.git/logs/refs/heads
/home/taponomareva/.password-store/study
/home/taponomareva/.password-store/bin
/home/taponomareva/.bashrc.d
```

Рис. 3.10: Вывод всех директорий в домашнем каталоге

4 Ответы на контрольные вопросы

1)Потоки ввода-вывода — это механизмы, с помощью которых происходит обмен данными между программами и устройствами. Основные потоки:

- Стандартный ввод (stdin) — поток, через который данные поступают в программу (например, с клавиатуры).
- Стандартный вывод (stdout) — поток, через который данные выводятся из программы (например, на экран).
- Стандартный вывод ошибок (stderr) — поток для вывода ошибок, который позволяет отделить сообщения об ошибках от основного вывода.

2)Разница между операцией > и »:

- `>` — оператор перенаправления вывода, который создает новый файл или перезаписывает существующий, если файл уже существует.
- `»` — оператор, который дописывает вывод в конец существующего файла, не перезаписывая его содержимое.

3)Конвейер — это механизм, позволяющий передавать вывод одной команды в качестве ввода для следующей команды. Он используется для последовательной обработки данных (например, `ps aux | grep "python"`).

4) Процесс — это выполняемая программа с выделенными системными ресурсами. Процесс отличается от программы тем, что программа — это набор

инструкций, а процесс — это экземпляр программы, выполняющийся в данный момент. Процесс имеет уникальный идентификатор (PID) и может быть многократным (запуск одной и той же программы).

5)PID и GID:

PID (Process ID) — уникальный идентификатор процесса. GID (Group ID) — идентификатор группы, к которой принадлежит процесс (например, для управления правами доступа).

6)Задачи — это отдельные процессы или потоки, которые выполняются системой. Для управления задачами используется команда `ps`, которая показывает список процессов, и команда `kill`, которая позволяет завершить процесс.

7)top и htop:

`top` — утилита для мониторинга процессов в реальном времени, которая отображает информацию о системе, таких как использование CPU, памяти, процессы и т. д. `htop` — более удобная версия `top`, с цветным интерфейсом и возможностью взаимодействовать с процессами, например, их завершать.

8) Команда поиска файлов — команда `find` позволяет искать файлы в файловой системе по различным критериям, например, по имени или времени модификации. Пример: `find /home -name "*.txt"` — найдет все файлы с расширением `.txt` в каталоге `/home`.

9) Для поиска файла по содержанию этого используют команду `grep`, которая ищет строки в файлах по заданному параметру. Пример: `grep -r "search_string" /path/to/directory` — ищет строки, содержащие `"search_string"` в файлах внутри указанного каталога.

10)Определение объема свободной памяти на жестком диске: используется команда `df` для получения информации о свободном и занятом пространстве на жестком диске. Пример: `df -h`

11)Определение объема домашнего каталога: используется команда `du` для подсчета использования пространства в каталоге. Пример: `du -sh /home/username`

- 12) Удаление зависшего процесса: выполняется команда `kill`, чтобы завершить зависший процесс. Пример: `kill -9` — где — идентификатор зависшего процесса, где параметр `-9` значит принудительное завершение процесса.

5 Выводы

Было произведено ознакомление с инструментами поиска файлов и фильтрации текстовых данных. Были приобретены практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

Список литературы

1. Курс на ТУИС