

Лабораторная работа №8

Дисциплина: Архитектура компьютера

Орлов Илья Сергеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	9
5	Контрольные вопросы	13
6	Выводы	16

Список иллюстраций

4.1	Запись содержимого каталога	9
4.2	Фильтрация конфигурационных файлов	10
4.3	Фильтрация поиска	10
4.4	Фильтрация поиска	11
4.5	Запись местоположений лог файлов	11
4.6	Завершение процессов через терминал	12
4.7	Вывод всех директорий	12

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 Задание

1. Осуществите вход в систему, используя соответствующее имя пользователя.
2. Запишите в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`.
Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге.
3. Выведите имена всех файлов из `file.txt`, имеющих расширение `.conf`, после чего запишите их в новый текстовый файл `conf.txt`.
4. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинающиеся с символа `s`? Предложите несколько вариантов, как это сделать.
5. Выведите на экран (по странично) имена файлов из каталога `/etc`, начинающиеся с символа `h`.
6. Запустите в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`.
7. Удалите файл `~/logfile`.
8. Запустите из консоли в фоновом режиме редактор `gedit`.
9. Определите идентификатор процесса `gedit`, используя команду `ps`, конвейер и фильтр `grep`. Как ещё можно определить идентификатор процесса?
10. Прочтите справку (`man`) команды `kill`, после чего используйте её для завершения процесса `gedit`.
11. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.
12. Воспользовавшись справкой команды `find`, выведите имена всех директорий, имеющих в вашем домашнем каталоге. Mar 4, 2025 feat(main): make

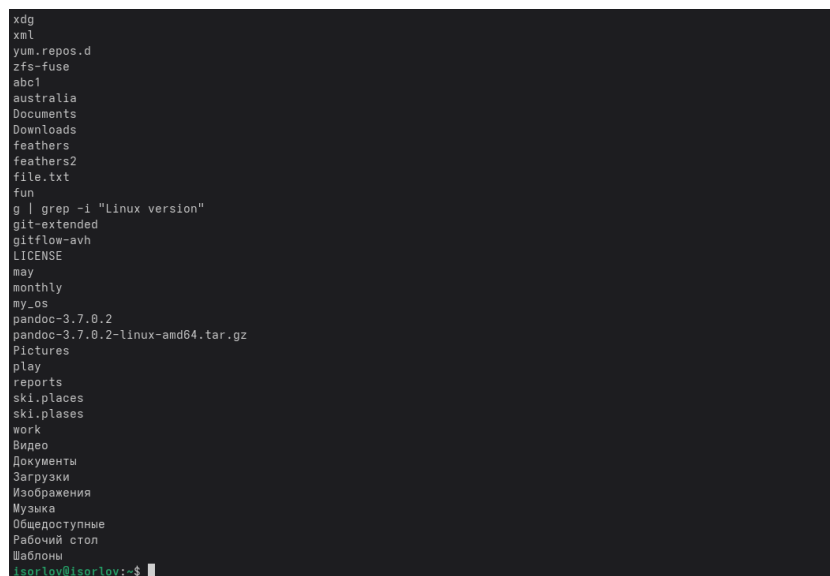
course structure

3 Теоретическое введение

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`. Например, команда `ls` выводит в стандартный поток вывода (консоль) список файлов в текущей директории. Потоки вывода и ввода можно перенаправлять на другие файлы или устройства. Проще всего это делается с помощью символов `>`, `>>`, `<`, `<<`. Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей. Команда `find` используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов.

4 Выполнение лабораторной работы

Записываю в файл содержимое каталога. (рис. 4.1)



```
xdg
xml
yum.repos.d
zfs-fuse
abc1
australia
Documents
Downloads
feathers
feathers2
file.txt
fun
g | grep -i "Linux version"
git-extended
gitflow-avh
LICENSE
may
monthly
my_os
pandoc-3.7.0.2
pandoc-3.7.0.2-linux-amd64.tar.gz
Pictures
play
reports
ski.places
ski.plases
work
Видео
Документы
Загрузки
Изображения
Музыка
Общедоступные
Рабочий стол
Шаблоны
Isorlov@Isorlov:~$
```

Рис. 4.1: Запись содержимого каталога

Записываю в файл все файлы конфигурации. (рис. 4.2)

```
abrt
adjtime
aliases
alsa
alternatives
anaconda
anacrontab
anthy-unicode.conf
asound.conf
at.deny
audit
authselect
avahi
bash_completion.d
bashrc
bindresvport.blacklist
binfmt.d
bluetooth
brlapi.key
brltty
brltty.conf
ceph
chkconfig.d
chromium
chrony.conf
cifs-utils
cockpit
colord
containers
credstore
credstore.encrypted
cron.d
cron.daily
cron.deny
cron.hourly
cron.monthly
```

Рис. 4.2: Фильтрация конфигурационных файлов

Вывожу все файлы, начинающиеся на с. (рис. 4.3)

```
isorlov@isorlov:~$ ls ~/c*
/home/isorlov/conf.txt
```

Рис. 4.3: Фильтрация поиска

Вывожу все файлы, начинающиеся на h. (рис. 4.4)

```
/etc/avahi/hosts
/etc/brltty/Contraction/ha.ctb
/etc/brltty/Input/bm/horizontal.kti
/etc/brltty/Input/hd
/etc/brltty/Input/hm
/etc/brltty/Input/ht
/etc/brltty/Input/hw
/etc/brltty/Text/he.ttb
/etc/brltty/Text/hi.ttb
/etc/brltty/Text/hr.ttb
/etc/brltty/Text/hu.ttb
/etc/brltty/Text/hy.ttb
/etc/containers/oci/hooks.d
/etc/firewalld/helpers
/etc/hp
/etc/hp/hplip.conf
/etc/httpd
/etc/httpd/conf/httpd.conf
/etc/libibverbs.d/hfi1verbs.driver
/etc/libibverbs.d/hns.driver
/etc/logrotate.d/httpd
/etc/nvme/hostnqn
/etc/nvme/hostid
/etc/sane.d/dll.d/hpaio
/etc/sane.d/hp.conf
/etc/sane.d/hp3900.conf
/etc/sane.d/hp4200.conf
/etc/sane.d/hp5400.conf
/etc/sane.d/hpsj5s.conf
/etc/sane.d/hs2p.conf
```

Рис. 4.4: Фильтрация поиска

Записываю все логи в файл. (рис. 4.5)

```
isorlov@isorlov:~$ find / -name "log*" > ~/logfile 2>/dev/null &
[1] 3180
isorlov@isorlov:~$ rm ~/logfile
[1]+  Выход 1          find / -name "log*" > ~/logfile 2> /dev/null
```

Рис. 4.5: Запись местоположений лог файлов

Открываю в фоне процесс и завершаю его. (рис. 4.6)

```
isorlov@isorlov:~$ gedit &  
[1] 3262  
isorlov@isorlov:~$ kill 3262
```

Рис. 4.6: Завершение процессов через терминал

Вывожу все директории домашнего каталога. (рис. 4.7)

```
isorlov@isorlov:~$ find ~ -type d  
/home/isorlov  
/home/isorlov/.mozilla  
/home/isorlov/.mozilla/extensions  
/home/isorlov/.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}  
/home/isorlov/.mozilla/plugins  
/home/isorlov/.mozilla/firefox  
/home/isorlov/.mozilla/firefox/Crash Reports  
/home/isorlov/.mozilla/firefox/Crash Reports/events  
/home/isorlov/.mozilla/firefox/Pending Pings  
/home/isorlov/.mozilla/firefox/cjd4eek8.default-release  
/home/isorlov/.mozilla/firefox/cjd4eek8.default-release/minidumps  
/home/isorlov/.mozilla/firefox/cjd4eek8.default-release/crashes  
/home/isorlov/.mozilla/firefox/cjd4eek8.default-release/crashes/events  
/home/isorlov/.mozilla/firefox/cjd4eek8.default-release/security_state
```

Рис. 4.7: Вывод всех директорий

5 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете? 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.
2. Объясните разницу между операцией `>` и `»`. Этот знак `>` - перенаправление ввода/вывода, а `»` - перенаправление в режиме добавления.
3. Что такое конвейер? Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.
4. Что такое процесс? Чем это понятие отличается от программы? Главное отличие между программой и процессом заключается в том, что программа - это набор инструкций, который позволяет ЦПУ выполнять определенную задачу, в то время как процесс - это исполняемая программа.
5. Что такое PID и GID? PPID - (`parent process ID`) идентификатор родительского процесса. Процесс может порождать и другие процессы. UID, GID - реальные идентификаторы пользователя и его группы, запустившего данный процесс.
6. Что такое задачи и какая команда позволяет ими управлять? Запущенные

фоном программы называются задачами (jobs). Ими можно управлять с помощью команды `jobs`, которая выводит список запущенных в данный момент задач.

7. Найдите информацию об утилитах `top` и `htop`. Каковы их функции? Команда `htop` похожа на команду `top` по выполняемой функции: они обе показывают информацию о процессах в реальном времени, выводят данные о потреблении системных ресурсов и позволяют искать, останавливать и управлять процессами.
8. У обеих команд есть свои преимущества. Например, в программе `htop` реализован очень удобный поиск по процессам, а также их фильтрация. В команде `top` это не так удобно — нужно знать кнопку для вывода функции поиска. Зато в `top` можно разделять область окна и выводить информацию о процессах в соответствии с разными настройками. В целом `top` намного более гибкая в настройке отображения процессов.
9. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды. Команда `find` - это одна из наиболее важных и часто используемых утилит системы Linux. Это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям. Утилита `find` предустановлена по умолчанию во всех Linux дистрибутивах, поэтому вам не нужно будет устанавливать никаких дополнительных пакетов. Это очень важная находка для тех, кто хочет использовать командную строку наиболее эффективно. Команда `find` имеет такой синтаксис: `find [папка] [параметры] критерий шаблон [действие]` Пример: `find /etc -name "p*" -print`
10. Можно ли по контексту (содержанию) найти файл? Если да, то как? `find / -type f -exec grep -H 'текстДляПоиска' {} ;`

11. Как определить объем свободной памяти на жёстком диске? С помощью команды `df -h`.
12. Как определить объем вашего домашнего каталога? С помощью команды `du -s`.
13. Как удалить зависший процесс? С помощью команды `kill %номер задачи`.

6 Выводы

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