

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ И ПРОГРАММНЫХ ТЕХНОЛОГИЙ

Отчёт по лабораторной работе

Курс: Операционные системы»

Тема: «Файловые системы»

Выполнил студент:

Бояркин Никита Сергеевич

Группа: 43501/3

Проверил:

Душутина Елена Владимировна

Санкт-Петербург
2016 г.

Содержание

1	Лабораторная работа №2	2
1.1	Характеристики системы	2
1.2	Ход работы	2
1.2.1	Фильтрация по одному примеру каждого типа файла	2
1.2.2	Получение всех жестких ссылок на файл	3
1.2.3	Анализ всех способов формирования ссылок	4
1.2.4	5
1.2.5	Утилита find	5
1.2.6	Утилиты od и *dump	6

Лабораторная работа №2

1.1 Характеристики системы

Некоторая информация об операционной системе и текущем пользователе:

```
nikita@nikita-pc:~$ who
nikita    tty7          2016-10-18 10:01 (:0)

nikita@nikita-pc:~$ cat /proc/version
Linux version 4.4.0-31-generic (buildd@lgw01-16) (gcc version 5.3.1 20160413 (Ubuntu
5.3.1-14ubuntu2.1) ) #50-Ubuntu SMP Wed Jul 13 00:07:12 UTC 2016
```

1.2 Ход работы

1.2.1 Фильтрация по одному примеру каждого типа файла

Решение в командной строке

Разработаем команду, которая выведет по одному примеру каждого типа файла из корневого каталога:

```
nikita@nikita-pc:~$ sudo ls / -l -R | awk '{if ($0~/^\/) path=substr($0, 0, length($0)-1); else { if ($0~/^1/) $(NF-2)=path/"$(NF-2); else {$NF=path/"$NF} print $0} }' |
grep -v ^/ | sort -k1.1,1.1 | uniq -w1
-r--r-- 1 root root 745 мар 30 2016 /etc/sudoers
brw-rw- 1 root disk 1, 0 окт 22 09:43 /dev/ram0
c----- 1 root root 5, 2 окт 22 09:43 /dev/pt/ptmx
drwx----- 2 root lp 4096 апр 21 2016 /etc/cup/ssl
lrwxrwxrwx 1 nikita nikita 10 июл 30 05:44 ControlPanel -> /home/nikita/files/software/
altera/13.1/quartus/linux/jre/bi/./jcontrol
total /104
```

Рассмотрим команду подробно:

- `ls / -l -R` - устанавливаем рекурсивный поиск по корневому каталогу с выводом полной информации.
- `awk` - скрипт, который добавляет полный путь в название файла.
- `if ($0~/^\/) path=substr($0, 0, length($0)-1);` - если строка начинается с `/` (каталог), то сохраняем текущий путь в переменную.
- `else { if ($0~/^1/) $(NF-2)=path/"$(NF-2);` - иначе, если это символическая ссылка (начинается с `l`), то изменяем путь в столбце `(NF-2)`.
- `else {$NF=path/"$NF} print $0} }` - иначе заменяем путь в последнем столбце `(NF)`.
- `grep -v ^/` - избавляемся от вывода каталогов.
- `sort -k1.1,1.1` - сортировка по первому символу.
- `uniq -w1` - уникальность по первому символу.

В результате работы команды были получены типы файлов с префиксами *-*, *b*, *c*, *d*, *l*, однако есть еще два типа файлов, их префиксы *p*, *s*. Рассмотрим каждый префикс подробнее:

- *-* - файл, обеспечивает хранение символьных и двоичных данных.
- *b* - блочное устройство, обеспечивает обращение к аппаратному обеспечению компьютера. Пример блочного устройства - жесткий диск.
- *c* - символьное устройство, обеспечивает обращение к аппаратному обеспечению компьютера. Пример символьного устройства - терминал.
- *d* - каталог, обеспечивает организацию доступа к файлам.
- *l* - символьная ссылка, обеспечивает предоставление доступа к файлам, расположенным на любых носителях.
- *p* - канал (FIFO), обеспечивает организацию взаимодействия процессов в операционной системе.
- *s* - сокет, обеспечивает организацию взаимодействия процессов в операционной системе.

Решение в виде *bash* скрипта

Решение аналогично предыдущему пункту, однако оформлено в виде *bash* скрипта. Отличие заключается в получении имени файла из аргументов командной строки и запись решения в этот файл:

```
#!/bin/bash

# Если пользователь указал имя результирующего файла, то используем его

filename=$1
if [ -z $filename ]; then
    # Если имя файла не указано, то используем имя по умолчанию
    filename="1.out"
fi

# Исполняем команду из предыдущего пункта, перенаправляя вывод в файл

ls / -l -R | awk '{if ($0~/^\/\//) path=substr($0, 0, length($0)-1); else { if($0~/^1/) $(NF-2)=path/"$(NF-2)"; else {NF=path/"$NF" print $0} }' | grep -v ^/ | sort -k1.1,1.1 | uniq -w1 > $filename
```

Запуск скрипта на исполнение происходит следующим образом:

```
nikita@nikita-pc:~/temp1$ sudo sh 1.sh filename
```

В папке со скриптом создался файл *filename*, в котором находится результат работы скрипта, аналогичный предыдущему пункту.

1.2.2 Получение всех жестких ссылок на файл

С помощью использования индексного дескриптора найдем все ссылки на указанный файл:

```
#!/bin/bash

# Если пользователь не указал имя файла, то выходим с ошибкой

filename=$1;
if [ -z $filename ]; then
    exit 1
fi

# Получим индексный дескриптор указанного файла, если не получилось, то выходим с ошибкой

inode="$(ls -li $filename | awk '{print $1}')"

if [ -z $inode ]; then
    exit 1
fi
```

Рекурсивно ищем все жесткие ссылки на индексный дескриптор в домашнем каталоге пользователя

```
ls $HOME -l -R -i | grep ^$inode
```

Результаты работы скрипта:

```
nikita@nikita-pc:~/temp1$ >tempfile
nikita@nikita-pc:~/temp1$ mkdir tempdir
nikita@nikita-pc:~/temp1$ ln tempfile tempdir/link_to_tempfile
nikita@nikita-pc:~/temp1$ ln tempfile link_to_tempfile
nikita@nikita-pc:~/temp1$ ls -l
total 20
(...)
-rw-rw-r-- 3 nikita nikita 0 окт 18 11:01 link_to_tempfile
drwxrwxr-x 2 nikita nikita 4096 окт 18 11:02 tempdir
-rw-rw-r-- 3 nikita nikita 0 окт 18 11:01 tempfile
nikita@nikita-pc:~/temp1$ sudo sh 2.sh tempfile
11278137 -rw-rw-r-- 3 nikita nikita 0 окт 18 11:01 link_to_tempfile
11278137 -rw-rw-r-- 3 nikita nikita 0 окт 18 11:01 tempfile
11278137 -rw-rw-r-- 3 nikita nikita 0 окт 18 11:01 link_to_tempfile
```

1.2.3 Анализ всех способов формирования ссылок

Рассмотрим действия команд *link*, *ln*, *ln -s*, *cp*. С помощью команды *ls -l* выясним какого рода объекты они порождают:

```
nikita@nikita-pc:~/temp1$ >tempfile
nikita@nikita-pc:~/temp1$ link tempfile templink
nikita@nikita-pc:~/temp1$ ls -l
(...)
-rw-rw-r-- 2 nikita nikita 0 окт 18 11:49 tempfile
-rw-rw-r-- 2 nikita nikita 0 окт 18 11:49 templink
nikita@nikita-pc:~/temp1$ rm templink tempfile

nikita@nikita-pc:~/temp1$ >tempfile
nikita@nikita-pc:~/temp1$ ln tempfile templink
nikita@nikita-pc:~/temp1$ ls -l
(...)
-rw-rw-r-- 2 nikita nikita 0 окт 18 11:49 tempfile
-rw-rw-r-- 2 nikita nikita 0 окт 18 11:49 templink
nikita@nikita-pc:~/temp1$ rm templink tempfile

nikita@nikita-pc:~/temp1$ >tempfile
nikita@nikita-pc:~/temp1$ ln -s tempfile templink
nikita@nikita-pc:~/temp1$ ls -l
(...)
-rw-rw-r-- 1 nikita nikita 0 окт 18 11:59 tempfile
lrwxrwxrwx 1 nikita nikita 8 окт 18 11:59 templink -> tempfile
nikita@nikita-pc:~/temp1$ rm templink tempfile

nikita@nikita-pc:~/temp1$ >tempfile
nikita@nikita-pc:~/temp1$ cp tempfile templink
nikita@nikita-pc:~/temp1$ ls -l
(...)
-rw-rw-r-- 1 nikita nikita 0 окт 18 12:02 tempfile
-rw-rw-r-- 1 nikita nikita 0 окт 18 12:02 templink
nikita@nikita-pc:~/temp1$ rm templink tempfile
```

Сделаем вывод о назначении команд *link*, *ln*, *ln -s*, *cp*:

- *link* - позволяет создавать только жесткие ссылки.
- *ln* - без ключей создает жесткую ссылку на файл.
- *ln -s* - с ключем *-s* создает символическую ссылку на файл.
- *cp* - создает новый файл.

Вывод всех символьных ссылок на файл

Напишем скрипт подсчитывающий все символьные ссылки на указанный файл:

```
#!/bin/bash

# Если пользователь не указал имя файла, то выходим с ошибкой

filename=$1;
if [ -z $filename ]; then
    exit 1
fi

# Рекурсивно ищем все символьные ссылки на файл в корневом каталоге

ls / -l -R | grep '\-> '$1
```

Результат работы скрипта:

```
nikita@nikita-pc:~/temp1$ >tempfile
nikita@nikita-pc:~/temp1$ ln -s tempfile dddd
nikita@nikita-pc:~/temp1$ ln -s tempfile gggg
nikita@nikita-pc:~/temp1$ ln -s tempfile yyyy
nikita@nikita-pc:~/temp1$ sudo sh 3.sh tempfile
lrwxrwxrwx 1 nikita nikita 8 окт 18 12:30 dddd -> tempfile
lrwxrwxrwx 1 nikita nikita 8 окт 18 12:30 gggg -> tempfile
lrwxrwxrwx 1 nikita nikita 8 окт 18 12:30 yyyy -> tempfile
```

1.2.4

1.2.5 Утилита find

find - утилита для поиска файлов по имени и другим свойствам в UNIX-подобных ОС. Может проводить поиск в одном или нескольких каталогах, с использованием критериев, заданных пользователем. По умолчанию возвращает все файлы в рабочей директории. также *find* позволяет применять действия ко всем найденным файлам.

Рассмотрим возможности команды *find* с несколькими ключами:

```
nikita@nikita-pc:~/temp1$ ls
0.log 1.log 1.sh 2.log 2.sh 3.log 3.sh 5.log
nikita@nikita-pc:~/temp1$ find -name "*.log"
./5.log
./2.log
./1.log
./3.log
./0.log
nikita@nikita-pc:~/temp1$ find -name "3*"
./3.sh
./3.log
nikita@nikita-pc:~/temp1$ ls ../temp
cat.txt      file_name  gdbtest    my_open.c  prog1.c    v1
designed.txt file_name.c gdbtest.c  pipe        res.txt     v2
eqv          foo.c      libfoo.so  pipe.c      result
example     foo.o      my_open    prog        script.sh
nikita@nikita-pc:~/temp1$ find ../temp -name "*.txt"
../temp/cat.txt
../temp/designed.txt
../temp/res.txt
nikita@nikita-pc:~/temp1$ find -size +500c
.
./2.sh
./5.log
./2.log
./3.log
./1.sh
nikita@nikita-pc:~/temp1$ find -size -500c
./1.log
```

```
./3.sh
./0.log
nikita@nikita-pc:~/temp1$ find -size -500c -exec ls -l {} \;
-rw-rw-r-- 1 nikita nikita 206 окт 18 10:28 ./1.log
-rw-rw-r-- 1 nikita nikita 360 окт 18 12:34 ./3.sh
-rw-rw-r-- 1 nikita nikita 257 окт 18 10:49 ./0.log
```

Рассмотрим результат исследования команды с несколькими ключами подробнее:

- *find -type* - осуществляет поиск по типу файла.
- *find -name* - осуществляет поиск файлов по имени, в основном используется для поиска по маске.
- *find -size* - осуществляет поиск файлов по размеру. Можно устанавливать нижнюю границу размера файла, верхнюю или обе вместе.
- *find -exec* - позволяет создавать вложенные команды. Аргумент "{}" заменяется на имя рассматриваемого файла, каждый раз, когда он встречается среди аргументов команды. Все символы за флагом *-exec* считаются ее аргументами до символа ";".

1.2.6 Утилиты *od* и **dump*

Рассмотрим команды *od* и *hexdump* с флагами *-c*, *-bc*:

```
nikita@nikita-pc:~/temp1$ echo "content">tempfile
nikita@nikita-pc:~/temp1$ ls -l tempfile
-rw-rw-r-- 1 nikita nikita 8 окт 18 12:55 tempfile
nikita@nikita-pc:~/temp1$ od tempfile
0000000 067543 072156 067145 005164
0000010
nikita@nikita-pc:~/temp1$ od -c tempfile
0000000  c  o  n  t  e  n  t  \n
0000010
nikita@nikita-pc:~/temp1$ od -bc tempfile
0000000 143 157 156 164 145 156 164 012
          c  o  n  t  e  n  t  \n
0000010
nikita@nikita-pc:~/temp1$ hexdump -v -e '"%2_ad | " 2 "%_c" "|\n"' tempfile
0 | co|
2 | nt|
4 | en|
6 | t\n|
```