

Липецкий государственный технический университет

Кафедра прикладной математики

Отчет по лабораторной работе № 3 «Процессы в операционной системе Linux» по курсу «ОС Linux»

Студент

подпись, дата

Гришагин Е.Е.
фамилия, инициалы

Группа

ПМ-19-2

Руководитель

ученая степень, ученое звание

подпись, дата

Кургасов В.В.
фамилия, инициалы

Липецк 2021 г.

Содержание

Задание кафедры	3
1. Ход работы	5
1.1. Часть I	5
1.1.1 Задание 1	5
1.1.2 Задание 2	5
1.1.3 Задание 3	7
1.2. Часть II	7
1.2.1 Задание 1	7
1.2.2 Задание 2	8
1.2.3 Задание 3	8
1.2.4 Задание 4	8
1.2.5 Задание 5	9
1.2.6 Задание 6	9
1.2.7 Задание 7	10
1.2.8 Задание 8	10
1.2.9 Задание 9	11
1.2.10 Задание 10	11
1.2.11 Задание 11	12
1.2.12 Задание 12	12
1.2.13 Задание 13	12
1.2.14 Задание 14	13
1.2.15 Задание 15	14
1.3. Часть III	14
1.3.1 Задание 1	14
1.3.2 Задание 2	15
1.3.3 Задание 3	15
1.3.4 Задание 4	16
2. Контрольные вопросы	19

Задание кафедры

Часть I

1. Получить следующую информацию о процессах текущего пользователя: идентификатор и имя владельца процесса, статус и приоритет процесса.
2. Завершить выполнение двух процессов, владельцем которых является текущий пользователь. Первый процесс завершить с помощью сигнала SIGINT, задав его имя, второй — с помощью сигнала SIGQUIT, задав его номер.
3. Определить идентификаторы и имена процессов, идентификатор группы которых не равен идентификатору группы текущего пользователя.

Часть II

1. Вывести информацию о текущем интерпретаторе команд
2. Вывести информацию о текущем пользователе
3. Вывести информацию о текущем каталоге
4. Вывести информацию об оперативной памяти и области подкачки
5. Вывести информацию о дисковой памяти
6. Получить идентификатор текущего процесса(PID)
7. Получить идентификатор родительского процесса(PPID)
8. Получить идентификатор процесса инициализации системы
9. Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд
10. Отобразить все процессы
11. Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе

12. Определить текущее значение `pid` по умолчанию
13. Запустить интерпретатор `bash` с понижением приоритета
14. Определить PID запущенного интерпретатора
15. Установить приоритет запущенного интерпретатора равным 5
16. Получить информацию о процессах `bash`

Часть III

1. Запустить в консоли на выполнение три задачи, две в интерактивном режиме, одну - в фоновом
2. Перевести одну из задач, выполняющихся в интерактивном режиме, в фоновый режим.
3. Провести эксперименты по переводу задач из фонового режима в интерактивный и наоборот.
4. Создать именованный канал для архивирования и осуществить передачу в канал списка файлов домашнего каталога вместе с подкаталогами (ключ `-R`) и одного каталога вместе с файлами и подкаталогами.

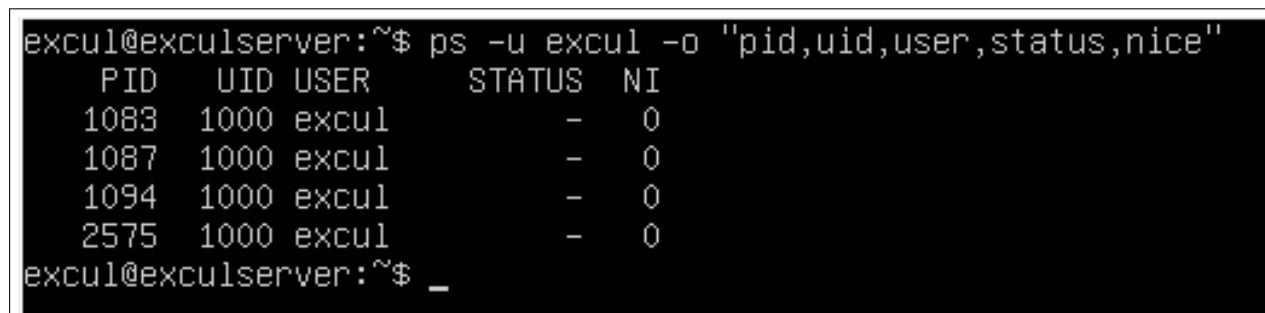
В отчете предоставьте все шаги ваших действий. То есть следует привести следующее: текст задания, а следом за ним снимок экрана консоли с результатами выполнения задания. Кроме того, перед скриншотом следует привести текстовую запись использованных команд. Кратко пояснить результаты выполнения всех команд

1. Ход работы

1.1. Часть I

1.1..1 Задание 1

Получить следующую информацию о процессах текущего пользователя: идентификатор и имя владельца процесса, статус и приоритет процесса.



```
excul@exculserver:~$ ps -u excul -o "pid,uid,user,status,nice"
  PID   UID  USER    STATUS  NI
  1083  1000  excul    -       0
  1087  1000  excul    -       0
  1094  1000  excul    -       0
  2575  1000  excul    -       0
excul@exculserver:~$ _
```

Рисунок 1 - Информация о процессах пользователя

1. ps -u excul -o "pid,uid,user,status,nice"

ps - утилита для вывода отчёта о процессах системы. В данном случае, использован параметр «-u», который выводит процессы по имени пользователя, если он указан, а если не указан, то пользователя вызвавшего команду. Далее, параметр «-o», который отвечает за форматирование строки.

1.1..2 Задание 2

Завершить выполнение двух процессов, владельцем которых является текущий пользователь. Первый процесс завершить с помощью сигнала SIGINT, задав его имя, второй — с помощью сигнала SIGQUIT, задав его номер.

```
excul@exculserver:~/123$ sh test.txt&
[1] 1434
excul@exculserver:~/123$ pkill -2 -u excul sh

excul@exculserver:~/123$ ^C
[1]+  Interrupt                  sh test.txt
excul@exculserver:~/123$ _
```

Рисунок 2.1. - Завершение с помощью SIGINT по имени

1. `sh test.txt&`
2. `pkill -2 -u excul sh`

Первая команда – это вызов сценарий «test.txt» в фоновом режиме, а вторая – это команда «pkill», которая позволяет завершать процессы с заданными паттернами. В данном случае в него добавлены параметры «-2», который сообщает с каким сигналом закончить процесс (в данном примере SIGINT), а «-u» указывает какому именно пользователю завершить процесс под именем «sh».

SIGINT – сигнал выхода из процесса по комбинации клавиш «Ctrl-C».

```
excul@exculserver:~/123$ sh test.txt&
[1] 1442
excul@exculserver:~/123$ kill -3 1442
excul@exculserver:~/123$ ^C
[1]+  Quit                        (core dumped) sh test.txt
excul@exculserver:~/123$ _
```

Рисунок 2.2. - Завершение с помощью SIGQUIT по номеру

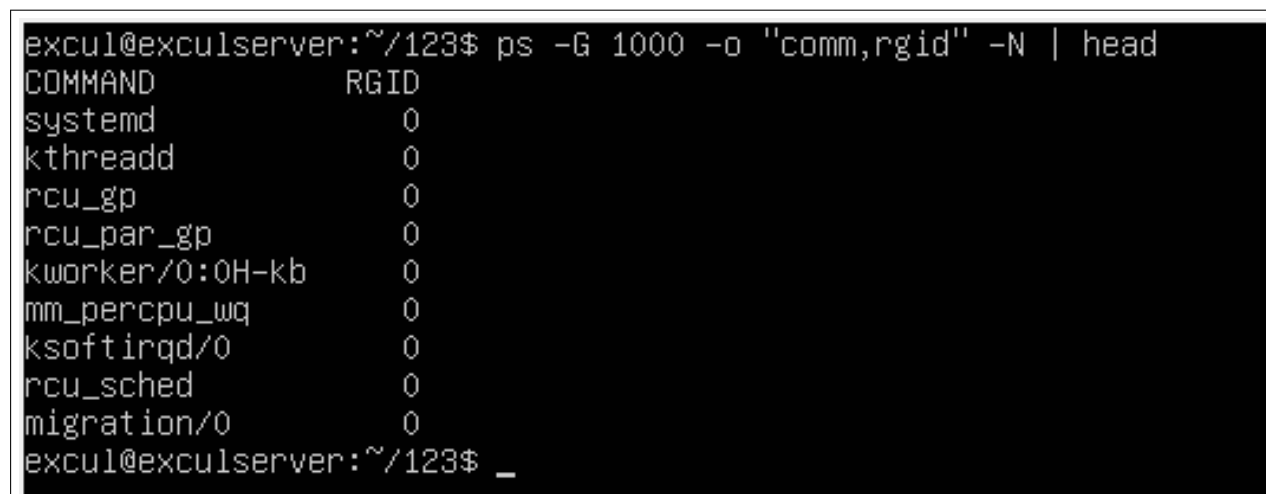
1. `sh test.txt&`
2. `kill -3 1442`

Первая команда уже описана выше. Вторая команда – kill позволяет завершить процесс по его номеру, а «-3», аналогично первому примеру сигнал SIGQUIT.

SIGQUIT – сигнал выхода из процесса с дампом памяти процесса.

1.1..3 Задание 3

Определить идентификаторы и имена процессов, идентификатор группы которых не равен идентификатору группы текущего пользователя.



```
excul@exculserver:~/123$ ps -G 1000 -o "comm,rgid" -N | head
COMMAND          RGID
systemd           0
kthreadd          0
rcu_gp            0
rcu_par_gp        0
kworker/0:0H-kb    0
mm_percpu_wq      0
ksoftirqd/0       0
rcu_sched          0
migration/0       0
excul@exculserver:~/123$ _
```

Рисунок 3 - Информация о процессах

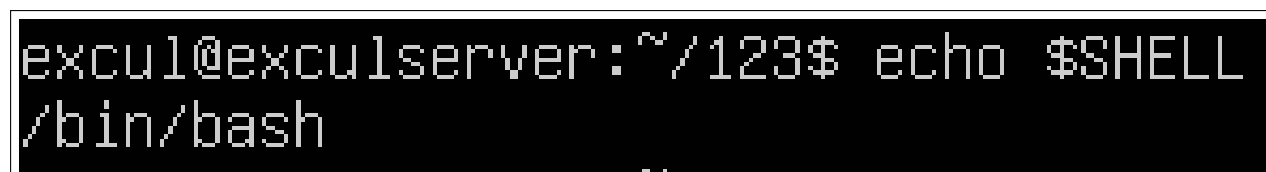
1. `ps -G 1000 -o "comm,rgid" -N | head`

В данном случае команда `ps` выполнена с параметром «-G», который считывает группу пользователей, которую нужно вывести, также в конце есть параметр «-N», который указывает на то, что вывод нужно инверсировать. Т.е. в нашем случае Должны вывестись все процессы системы, группа пользователей которых не имеет `id = 1000`. Ещё использована команда `head` через канал, которая позволяет вывести только «верхушку» отчёта `ps`.

1.2. Часть II

1.2..1 Задание 1

Вывести информацию о текущем интерпретаторе команд



```
excul@exculserver:~/123$ echo $SHELL
/bin/bash
```

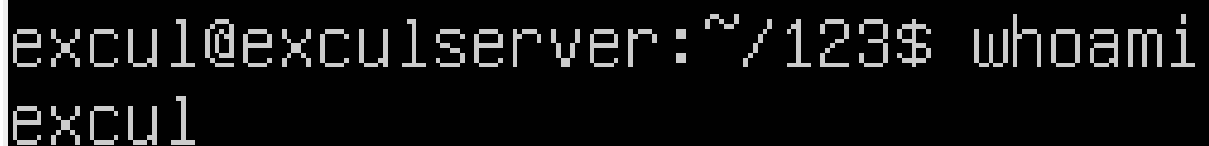
Рисунок 4 - Информация об интерпритаторе команд

1. `echo $SHELL`

Команда `echo` выводит какие-то данные в терминал. В данном случае, переменная окружения `$SHELL`, которая выводит путь к интерпритатору.

1.2..2 Задание 2

Вывести информацию о текущем пользователе



```
excul@exculserver:~/123$ whoami
excul
```

Рисунок 5 - Имя текущего пользователя

1. `whoami`

`whoami` выводит имя текущего пользователя

1.2..3 Задание 3

Вывести информацию о текущем каталоге



```
excul@exculserver:~/123$ pwd
/home/excul/123
excul@exculserver:~/123$
```

Рисунок 6 - Текущий каталог

1. `pwd`

`pwd` выводит путь к текущему каталогу

1.2..4 Задание 4

Вывести информацию об оперативной памяти и области подкачки


```
excul@exculserver:~/123$ free
              total        used        free      shared  buff/cache   available
Mem:      2035456      143360      1417620         1028       474476      1734452
Swap:      1261564           0       1261564
excul@exculserver:~/123$
```

Рисунок 7 - Информация об оперативной памяти

1. free

free выводит информацию об оперативной памяти

1.2..5 Задание 5

Вывести информацию о дисковой памяти

```
excul@exculserver:~/123$ df
Filesystem            1K-blocks    Used Available Use% Mounted on
udev                  972560         0    972560   0% /dev
tmpfs                 203548      1028    202520   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 7155192 4445060    2326952  66% /
tmpfs                1017728         0    1017728   0% /dev/shm
tmpfs                  5120         0       5120   0% /run/lock
tmpfs                1017728         0    1017728   0% /sys/fs/cgroup
/dev/loop2             63360      63360         0 100% /snap/core20/1169
/dev/loop3             63360      63360         0 100% /snap/core20/1242
/dev/loop4             68864      68864         0 100% /snap/lxd/21545
/dev/loop6             33280      33280         0 100% /snap/snapd/13640
/dev/loop0             56832      56832         0 100% /snap/core18/2246
/dev/loop7             33280      33280         0 100% /snap/snapd/13270
/dev/loop5             68864      68864         0 100% /snap/lxd/21835
/dev/sda2             999320    207404     723104  23% /boot
tmpfs                 203544         0    203544   0% /run/user/1000
/dev/loop8             56832      56832         0 100% /snap/core18/2253
excul@exculserver:~/123$ _
```

Рисунок 8 - Информация о дисковой памяти

1. df

df выводит информацию о дисковой памяти

1.2..6 Задание 6

Получить идентификатор текущего процесса(PID)

```
excul@exculserver:~/123$ echo $$
961
```

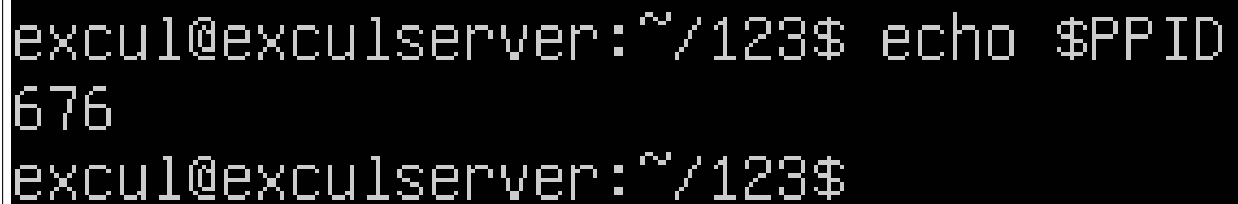
Рисунок 9 - PID текущего процесса

1. echo \$\$

Аналогично первому заданию

1.2..7 Задание 7

Получить идентификатор родительского процесса(PPID)



```
excul@exculserver:~/123$ echo $PPID
676
excul@exculserver:~/123$
```

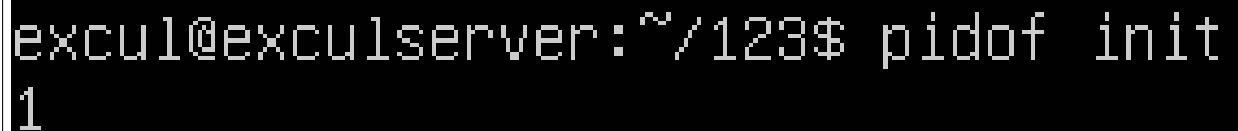
Рисунок 10 - PPID текущего процесса

1. echo \$\$PPID

Аналогично первому заданию

1.2..8 Задание 8

Получить идентификатор процесса инициализации системы



```
excul@exculserver:~/123$ pidof init
1
```

Рисунок 11 - PID init

1. pidof init

pidof позволяет получить PID по имени процесса

1.2..9 Задание 9

Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд

```
excul@exculserver:~/123$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
excul      961  0.0  0.2   8520   5720 tty1      S   17:01   0:00 -bash
excul     1577  0.0  0.1   8892   3468 tty1      R+  19:54   0:00 ps -u
excul@exculserver:~/123$
```

Рисунок 12 - Информация о процессах текущего пользователя

1. `ps -u`

Описание команды `ps` есть в первой части.

1.2..10 Задание 10

Отобразить все процессы

```
excul@exculserver:~/123$ ps -e | head
      PID TTY          TIME CMD
      1 ?           00:00:02 systemd
      2 ?           00:00:00 kthreadd
      3 ?           00:00:00 rcu_gp
      4 ?           00:00:00 rcu_par_gp
      6 ?           00:00:00 kworker/0:0H-kblockd
      8 ?           00:00:00 mm_percpu_wq
      9 ?           00:00:00 ksoftirqd/0
     10 ?           00:00:02 rcu_sched
     11 ?           00:00:00 migration/0
excul@exculserver:~/123$ _
```

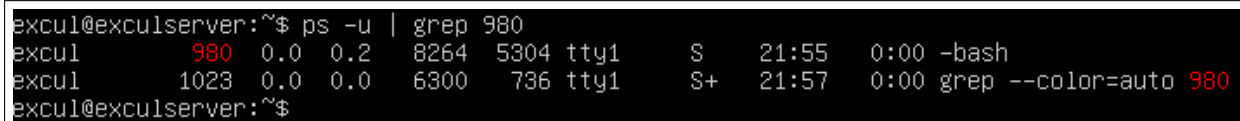
Рисунок 13 - Все процессы

1. `ps -e | head`

Параметр «-e» команды `ps` выводит вообще все процессы запущенные в системе

1.2..11 Задание 11

Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе



```
excul@exculserver:~$ ps -u | grep 980
excul      980  0.0  0.2  8264  5304 tty1      S   21:55   0:00 -bash
excul     1023  0.0  0.0   6300    736 tty1      S+  21:57   0:00 grep --color=auto 980
excul@exculserver:~$
```

Рисунок 14 - Все процессоры пользователя в текущем интерпритаторе

1. `ps -u | grep 890`

grep описана во 2 лабораторной и находит все совпадения PID для команды `ps -u`.

1.2..12 Задание 12

Определить текущее значение `nice` по умолчанию



```
excul@exculserver:~/123$ nice
0
excul@exculserver:~/123$
```

Рисунок 15 - Текущее значение `nice` по умолчанию

1. `nice`

`nice` выводит текущее значение приоритета по умолчанию

1.2..13 Задание 13

Запустить интерпретатор `bash` с понижением приоритета

```
excul@exculserver:~/123$ nice -n 10 bash
excul@exculserver:~/123$ ps -u excul -o "user,comm,pid,nice"
USER      COMMAND      PID  NI
excul     bash         961   0
excul     bash        1621  10
excul     ps           1628  10
excul@exculserver:~/123$ _
```

Рисунок 16 - Запуск bash и проверка

1. `nice -n 10 bash`
2. `ps -u excul - o "user,comm,pid,nice"`

«-n» в параметрах команды `nice` сигнализирует о том на какое количество единиц будет повышен приоритет запущенного процесса.

1.2..14 Задание 14

Определить PID запущенного интерпретатора и установить его приоритет равным 5

```
excul@exculserver:~$ echo $$
1781
excul@exculserver:~$ renice -n 5 1781
1781 (process ID) old priority 0, new priority 5
excul@exculserver:~$
```

Рисунок 17 - Установка приоритета у запущенного приложения

1. `echo $$`
2. `renice -n 5 1781`

«-n» аналогичен и для команды `renice`, которая принимает для `pid` для определения процесса.

1.2..15 Задание 15

Получить информацию о процессах bash

```
excul@exculserver:~$ echo $$
1781
excul@exculserver:~$ renice -n 5 1781
1781 (process ID) old priority 0, new priority 5
excul@exculserver:~$ ps lax | grep bash
4 1000 1781 1631 25 5 8264 5284 do_wai SN tty1 0:00 -bash
0 1000 1805 1781 25 5 6300 736 - RN+ tty1 0:00 grep --color=auto bash
excul@exculserver:~$ _
```

Рисунок 18 - Информация о процессах bash

1. `ps lax | grep bash`

«-n» аналогичен и для команды `renice`, которая принимает для `pid` для определения процесса.

1.3. Часть III

1.3..1 Задание 1

Запустить в консоли на выполнение три задачи, две в интерактивном режиме, одну - в фоновом.

```
excul@exculserver:~/123$ sh test.txt&
[1] 1823
excul@exculserver:~/123$ sh test.txt&
[2] 1824
excul@exculserver:~/123$ sh test.txt
```

Рисунок 19 - Запуск задач

1. `sh test.txt&`

2. `sh test.txt&`

3. `sh test.txt`

& говорит интерпретатору о том, что процесс запускается в фоновом режиме, если без него, то в интерактивном.

1.3..2 Задание 2

Перевести одну из задач, выполняющихся в интерактивном режиме, в фоновый режим.

```
excul@exculserver:~/123$ sh test.txt&
[1] 1835
excul@exculserver:~/123$ sh test.txt&
[2] 1836
excul@exculserver:~/123$ sh test.txt
^Z
[3]+  Stopped                  sh test.txt
excul@exculserver:~/123$ jobs
[1]  Running                    sh test.txt &
[2]- Running                    sh test.txt &
[3]+  Stopped                  sh test.txt
excul@exculserver:~/123$ bg 3
[3]+ sh test.txt &
excul@exculserver:~/123$ jobs
[1]  Running                    sh test.txt &
[2]- Running                    sh test.txt &
[3]+ Running                    sh test.txt &
excul@exculserver:~/123$
```

Рисунок 20 - Перевод задачи в фоновый режим и проверка

1. jobs

2. bg 3

3. jobs

jobs выводит все фоновые и остановленные интерактивные задачи. Оттуда мы узнаём jid интерактивного процесса и преводим его в фон командой bg «jid»

1.3..3 Задание 3

Провести эксперименты по переводу задач из фонового режима в интерактивный и наоборот.

```
excul@exculserver:~/123$ jobs
[1]  Running                  sh test.txt &
[2]-  Running                  sh test.txt &
[3]+  Running                  sh test.txt &
excul@exculserver:~/123$ fg 1
sh test.txt
^Z
[1]+  Stopped                  sh test.txt
excul@exculserver:~/123$ fg 2
sh test.txt
^Z
[2]+  Stopped                  sh test.txt
excul@exculserver:~/123$ bg 2
[2]+ sh test.txt &
excul@exculserver:~/123$
```

Рисунок 21 - Эксперементы

1. jobs
2. fg 1
3. fg 2
4. fg 2

Эксперементы с переводом с помощью команд fg и bg.

1.3..4 Задание 4

Создать именованный канал для архивирования и осуществить передачу в канал списка файлов домашнего каталога вместе с подкаталогами (ключ -R) и одного каталога вместе с файлами и подкаталогами.


```

excul@exculserver:~$ mkfifo pipe
excul@exculserver:~$ ls -R /home > pipe&
[3] 2309
excul@exculserver:~$ gzip -c pipe > arch.gz&
[4] 2310
excul@exculserver:~$ zcat arch.gz
/home:
excul
user

/home/excul:
123
arch.gz
pipe

/home/excul/123:
test2.txt
test.txt

/home/user:
1.txt
3.txt
hl
sl
[3]-  Done                  ls --color=auto -R /home > pipe
[4]+  Done                  gzip -c pipe > arch.gz
excul@exculserver:~$ tar -zcf arch.tar.gz arch.gz&
[3] 2368
excul@exculserver:~$ ls
123  arch.gz  arch.tar.gz  pipe
[3]+  Done                  tar -zcf arch.tar.gz arch.gz
excul@exculserver:~$ tar -tvf arch.tar.gz
-rw-rw-r-- excul/excul    103 2021-11-21 22:55 arch.gz
excul@exculserver:~$ _

```

Рисунок 21 - Архивирование при помощи именного канала

1. `mkfifo pipe`
2. `ls -R /home > pipe&`
3. `gzip -c pipe > arch.gz&`
4. `zcat arch.gz`
5. `tar -zcf arch.tar.gz arch.gz&`
6. `ls`
7. `tar -tvf arch.tar.gz`

Командой `mkfifo` я создаю именованный канал с именем `pipe`

Далее командой `ls` с параметром «-R», которое сигнализирует о рекурсивном выводе поданной папки, я перенаправляю вывод в именованный канал `pipe`

Далее командой `gzip` с параметром «-c», который позволяет выводит архив в стандартный ввод, я создаю архив с сжатой копией папки `/home` при помощи перенаправления вывода из именованного канала.

Далее я проверяю содержимое архива командой `zcat`

Далее командой `tar` я создаю архив `arch.tar.gz` при помощи параметров `zcf`, где `z` отвечает за архивирование `tar`-архива в `gzip`, `c` за создание файла, `f` за запись в архив.

Далее командой `ls` я проверяю создание архивов.

И в конце командой `tar` проверяю содержимое при помощи параметров `tvf`.

2. Контрольные вопросы

1. Вопрос: Перечислите состояния задачи в ОС Ubuntu.

Ответ: running (выполнения) , sleeping (спячки), останов (stopped), zombie (зомби) , dead (смерти), active (активный), expired (неактивный) .

2. Вопрос: Как создаются задачи задачи в ОС Ubuntu?

Ответ: Задачи создаются системной функцией clone.

3. Вопрос: Назовите классы потоков ОС Ubuntu

Ответ: Потоки реального времени, обслуживаемые по алгоритму FIFO, Потоки реального времени, обслуживаемые в порядке циклической очереди, Потоки разделения времени.

4. Вопрос: Как используется приоритет планирования при запуске задачи

Ответ: Чем выше приоритет, тем меньше должно быть время отклика и тем большая доля процессорного времени достанется процессу.

5. Вопрос: Как можно изменить приоритет для выполняющейся задачи?

Ответ: Командой renice.