



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет имени
Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №3 по курсу «Операционные системы»

Тема Основы Linux

Студент Богаченко А. Е.

Группа ИУ7-56Б

Оценка (баллы) _____

Преподаватели Рязанова Н.Ю.

Задание 1

Изучение команд Shell:

- используя команду `mkdir` создайте директорию именем своей группы;
- перейдите в созданную директорию с помощью команды `cd`;
- создайте поддиректорию, например, используя свою фамилию;
- команда `ls`;
- команда `ps`;

```
(root👤NebuchadnezzaR)-[~]
# mkdir iu7 && cd iu7

(root👤NebuchadnezzaR)-[~/iu7]
# mkdir bogachenko && cd bogachenko

(root👤NebuchadnezzaR)-[~/iu7/bogachenko]
# ls -la
total 8
drwxr-xr-x 2 root root 4096 Dec 19 02:07 .
drwxr-xr-x 3 root root 4096 Dec 19 02:07 ..

(root👤NebuchadnezzaR)-[~/iu7/bogachenko]
# ps
  PID TTY          TIME CMD
 261016 pts/5        00:00:00 zsh
1074583 pts/5        00:00:00 sudo
1074584 pts/5        00:00:01 zsh
1078626 pts/5        00:00:00 ps

(root👤NebuchadnezzaR)-[~/iu7/bogachenko]
# pwd
/root/iu7/bogachenko
```

Рисунок 1 – Создание директории, поддиректории, `cd`, `ps`, `aux`

Задание 2

Процессы:

- напишите программу, в которой создается дочерний процесс и организуйте как в предке, так и в потомке бесконечные циклы, в которых выводятся идентификаторы процессов с помощью системного вызова `getpid()`
- запустите программу и посмотрите идентификаторы созданных процессов: предка и потомка;
- для получения процесса зомби выполните следующие действия: а) удалите командой `kill` потомка и посмотрите с помощью команды `ps` его новый статус – Z; б) удалите предка;
- для получения “осиротевшего” процесса запустите программу еще раз, но в этот раз удалите предка и посмотрите с помощью команды `ps` идентификатор предка у продолжающего выполняться потомка – идентификатор предка будет изменен на 1, так как процесс был “усыновлен” процессом с идентификатором 1 процессом “открывшим” терминал в случае, если используется Unix BSD, или идентификатор процессов-посредников в случае, Linux Ubuntu.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 int main(void) {
5     int childpid;
6     if ((childpid = fork()) == -1) {
7         perror("Can't fork.\n");
8         return EXIT_FAILURE;
9     } else if (childpid == 0) {
10        while (1) {
11            printf("child pid = %d\n", getpid());
12        }
13        return EXIT_SUCCESS;
14    } else {
15        while (1) {
16            printf("parent pid = %d\n", getpid());
17        }
18        return 0;
19    }
```

```
19 }
20 }
```

```
parent pid = 1080563
child pid = 1080564
parent pid = 1080563
child pid = 1080564
parent pid = 1080563
child pid = 1080564
parent pid = 1080563
child pid = 1080564
parent pid = 1080563
child pid = 1080564
```

Рисунок 2 – Вывод программы

```
(root👤NebuchadnezzaR)-[~/iu7/bogachenko]
# ps aux | grep a.out
root    1080563 53.6  0.0   2304    548 pts/7    S+   03:20   0:03 ./a.out
root    1080564 53.5  0.0   2304     68 pts/7    S+   03:20   0:03 ./a.out
root    1080567  0.0  0.0   6256   2316 pts/8    S+   03:20   0:00 grep --color=auto a.out

(root👤NebuchadnezzaR)-[~/iu7/bogachenko]
# kill 1080564

(root👤NebuchadnezzaR)-[~/iu7/bogachenko]
# ps aux | grep a.out
root    1080563 58.1  0.0   2304    548 pts/7    S+   03:20   0:10 ./a.out
root    1080564 52.5  0.0      0      0 pts/7    Z+   03:20   0:09 [a.out] <defunct>
root    1080582  0.0  0.0   6256   2316 pts/8    S+   03:20   0:00 grep --color=auto a.out

(root👤NebuchadnezzaR)-[~/iu7/bogachenko]
# kill 1080563

(root👤NebuchadnezzaR)-[~/iu7/bogachenko]
# ps aux | grep a.out
root    1080587  0.0  0.0   6256   2204 pts/8    S+   03:20   0:00 grep --color=auto a.out
```

Рисунок 3 – Создание процесса zombie

Задание 3

Продемонстрировать работу pipe:

- создание и запись в pipe;
- чтение из pipe.

```
(root👤NebuchadnezzaR)-[~/iu7/bogachenko]  
# mknod pipe p && ls  
pipe  
  
(root👤NebuchadnezzaR)-[~/iu7/bogachenko]  
# echo "Hello world >" > pipe
```

Рисунок 4 – Создание и запись в pipe

```
(root👤NebuchadnezzaR)-[~/iu7/bogachenko]  
# tee < pipe  
Hello world >
```

Рисунок 5 – Чтение из pipe

Задание 4

Изучение `softlink` и `hardlink`:

- создать `hardlink`;
- создать и изменить `softlink`.

```
(root@NebuchadnezzaR)-[~/iu7/bogachenko]
# ls -li
total 4
3808139 -rw-r--r-- 1 root root 331 Dec 19 02:34 fork.c
3808138 prw-r--r-- 1 root root  0 Dec 19 02:26 pipe

(root@NebuchadnezzaR)-[~/iu7/bogachenko]
# ln fork.c hardlink && ls -li
total 8
3808139 -rw-r--r-- 2 root root 331 Dec 19 02:34 fork.c
3808139 -rw-r--r-- 2 root root 331 Dec 19 02:34 hardlink
3808138 prw-r--r-- 1 root root  0 Dec 19 02:26 pipe

(root@NebuchadnezzaR)-[~/iu7/bogachenko]
# rm hardlink && ls -li
total 4
3808139 -rw-r--r-- 1 root root 331 Dec 19 02:34 fork.c
3808138 prw-r--r-- 1 root root  0 Dec 19 02:26 pipe
```

Рисунок 6 – Создание `hardlink`

```
(root@NebuchadnezzaR)-[~/iu7/bogachenko]
# ls -li
total 4
3808139 -rw-r--r-- 1 root root 331 Dec 19 02:34 fork.c
3808138 prw-r--r-- 1 root root  0 Dec 19 02:26 pipe

(root@NebuchadnezzaR)-[~/iu7/bogachenko]
# ln -s fork.c softlink && ls -li
total 4
3808139 -rw-r--r-- 1 root root 331 Dec 19 02:34 fork.c
3808138 prw-r--r-- 1 root root  0 Dec 19 02:26 pipe
3808140 lrwxrwxrwx 1 root root  6 Dec 19 02:42 softlink -> fork.c

(root@NebuchadnezzaR)-[~/iu7/bogachenko]
# ln -sf ~/root/ ./softlink && ls -li
total 4
3808139 -rw-r--r-- 1 root root 331 Dec 19 02:34 fork.c
3808138 prw-r--r-- 1 root root  0 Dec 19 02:26 pipe
3808141 lrwxrwxrwx 1 root root 11 Dec 19 02:42 softlink -> /root/root/
```

Рисунок 7 – Создание и изменение `softlink`