



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

Студент _____ Топорков Павел _____

Группа _____ ИУ7-53Б _____

Дисциплина _____ Операционные системы _____

Преподаватель: _____ Рязанова Н. Ю. _____

подпись, дата

Фамилия, И.О.

Оценка _____

Москва — 2020 г.

Оглавление

1	Изучение команд Shell	2
1.1	Знакомство с командами mkdir и cd	2
1.2	Знакомство с командами ls и ps	3
2	Зомби и осиротевшие процессы	4
3	Жесткие и гибкие ссылки на файлы	7
4	Программные каналы	8
5	Приоритеты процессов	9

1 Изучение команд Shell

1.1 Знакомство с командами mkdir и cd

Используя команду `mkdir` создайте директорию именем своей группы. Перейдите в созданную директорию с помощью команды `cd`. Создайте поддиректорию, например, используя свою фамилию.



```
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03$ mkdir iu7-53b
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03$ cd iu7-53b
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b$ mkdir toporkov
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b$ cd toporkov
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$
```

Рисунок 1.1: Создание деректорий и переход по ним.

1.2 Знакомство с командами ls и ps

Команда ls выводит список файлов. Например: ls / - выводит список файлов из root.

```
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$ ls /  
bin  cdrom  etc  lib  lib64  lost+found  mnt  proc  run  snap  swapfile  tmp  var  
boot  dev  home  lib32  libx32  media  opt  root  sbin  srv  sys  usr  
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$
```

Рисунок 1.2: Выполнение команды "ls /"

Команда ps выводит список процессов в системе.

```
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$ ps al  
F  UID      PID    PPID  PRI   NI     VSZ   RSS WCHAN  STAT TTY        TIME COMMAND  
4  125     50114   50060   20    0  172628  6704 -        Ssl+  tty1        0:00 /usr/lib/gdm3/gdm-x-se  
4  125     50116   50114   20    0  116720  16796 -        Sl+   tty1        0:06 /usr/lib/xorg/Xorg vt1  
0  125     50182   50114   20    0  271040  14080 -        Sl+   tty1        0:00 /usr/libexec/gnome-ses  
4  1000    51975   51953   20    0  172628  6368 poll_s  Ssl+  tty2        0:00 /usr/lib/gdm3/gdm-x-se  
4  1000    51977   51975   20    0  146456  38788 ep_pol  Sl+   tty2        6:29 /usr/lib/xorg/Xorg vt2  
0  1000    51995   51975   20    0  197024  14040 poll_s  Sl+   tty2        0:00 /usr/libexec/gnome-ses  
0  1000   155076  155051   20    0   19620   5404 do_wai  Ss    pts/0       0:00 bash  
0  1000   155994  155076   20    0   20028   3152 -       R+    pts/0       0:00 ps al  
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$
```

Рисунок 1.3: Выполнение команды "ps al"

2 Зомби и осиротевшие процессы

Задание: 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()`.

1. Запустите программу и посмотрите идентификаторы созданных процессов: предка и потомка.

2. Для получения процесса зомби выполните следующие действия: · удалите командой `kill` потомка и посмотрите с помощью команды `ps` его новый статус – `Z`; · удалите предка.

3. 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.

```
#include <stdio.h>
int main()
{
    int childpid;
    if ((childpid = fork())== -1)
    {
        perror("Can't fork.\n");
        return 1;
    }
    else if (childpid == 0)
    {
        while (1) printf(" %d ", getpid());
        return 0;
    }
    else
    {
        while(1) printf(" %d ",getpid());
        return 0;
    }
}
```

Рисунок 2.1: Код программы.

```

7624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624
157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 15
624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624
157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157
24 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624
157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 1576
4 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624
157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 15762
157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 1
157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624
157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 15
624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624
157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157
24 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624
157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 1576
4 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624
157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 15762
157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 1
7624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624
157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 15
624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624
157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157
24 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624
157624 157624 157624 157624 157624 157624 157624 157624 157624 157624 157624

```

Рисунок 2.2: Вывод программы.

```

parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03$ ps -al
F S  UID      PID      PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   125     50116    50114  0  80   0 - 29180 -      tty1         00:00:06 Xorg
0 S   125     50182    50114  0  80   0 - 67760 -      tty1         00:00:00 gnome-sess
4 S  1000     51977    51975  0  80   0 - 37638 ep_pol tty2         00:06:45 Xorg
0 S  1000     51995    51975  0  80   0 - 49256 poll_s tty2         00:00:00 gnome-sess
0 R  1000    157623   155076 23  80   0 - 622 -      pts/0        00:00:14 a.out
1 R  1000    157624   157623 23  80   0 - 622 -      pts/0        00:00:14 a.out
0 R  1000    157802   157751  0  80   0 - 5007 -      pts/1        00:00:00 ps
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03$ kill 157624
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03$ ps -al
F S  UID      PID      PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   125     50116    50114  0  80   0 - 29180 -      tty1         00:00:06 Xorg
0 S   125     50182    50114  0  80   0 - 67760 -      tty1         00:00:00 gnome-sess
4 S  1000     51977    51975  0  80   0 - 37638 ep_pol tty2         00:06:56 Xorg
0 S  1000     51995    51975  0  80   0 - 49256 poll_s tty2         00:00:00 gnome-sess
0 R  1000    157623   155076 26  80   0 - 622 -      pts/0        00:00:57 a.out
1 Z  1000    157624   157623 20  80   0 - 0 -      pts/0        00:00:45
0 R  1000    158235   157751  0  80   0 - 5007 -      pts/1        00:00:00 ps
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03$

```

Рисунок 2.3: Создание процесса зомби.

Снова запустим программу.

```
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03$ ps -al
F S      UID        PID      PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S      125        50116     50114  0  80   0 - 29180 -        tty1        00:00:06 Xorg
0 S      125        50182     50114  0  80   0 - 67760 -        tty1        00:00:00 gnome-session-b
4 S     1000        51977     51975  0  80   0 - 38466 ep_pol tty2        00:07:10 Xorg
0 S     1000        51995     51975  0  80   0 - 49256 poll_s tty2        00:00:00 gnome-session-b
0 R     1000       158801    155076 21  80   0 -    622 -        pts/0       00:00:01 a.out
1 R     1000       158802    158801 20  80   0 -    622 -        pts/0       00:00:01 a.out
0 R     1000       158819    157751  0  80   0 -    5007 -        pts/1       00:00:00 ps
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03$ kill 158801
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03$ ps -al
F S      UID        PID      PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S      125        50116     50114  0  80   0 - 29180 -        tty1        00:00:06 Xorg
0 S      125        50182     50114  0  80   0 - 67760 -        tty1        00:00:00 gnome-session-b
4 S     1000        51977     51975  0  80   0 - 38466 ep_pol tty2        00:07:15 Xorg
0 S     1000        51995     51975  0  80   0 - 49256 poll_s tty2        00:00:00 gnome-session-b
1 R     1000       158802     1980 24  80   0 -    622 -        pts/0       00:00:18 a.out
0 R     1000       158986    157751  0  80   0 -    5007 -        pts/1       00:00:00 ps
```

Рисунок 2.4: Создание осиротевшего процесса.

3 Жесткие и гибкие ссылки на файлы

Задание: введите команду `ls` и сделайте `link` на файл и затем `ls`. После этого уничтожьте созданную ссылку и вызовите `ls`.

```
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$ ls -il
total 24
2232866 -rwxrwxr-x 1 parallels parallels 16816 Dec 14 16:02 a.out
2233489 -rw-rw-r-- 1 parallels parallels 347 Dec 14 16:01 main.c
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$ ln main.c newlink
link
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$ ls -il
total 28
2232866 -rwxrwxr-x 1 parallels parallels 16816 Dec 14 16:02 a.out
2233489 -rw-rw-r-- 2 parallels parallels 347 Dec 14 16:01 main.c
2233489 -rw-rw-r-- 2 parallels parallels 347 Dec 14 16:01 newlink
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$ rm newlink
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$ ls -il
total 24
2232866 -rwxrwxr-x 1 parallels parallels 16816 Dec 14 16:02 a.out
2233489 -rw-rw-r-- 1 parallels parallels 347 Dec 14 16:01 main.c
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$
```

Рисунок 3.1: Создание и удаление жесткой ссылки.

Задание: создайте новую символическую ссылку.

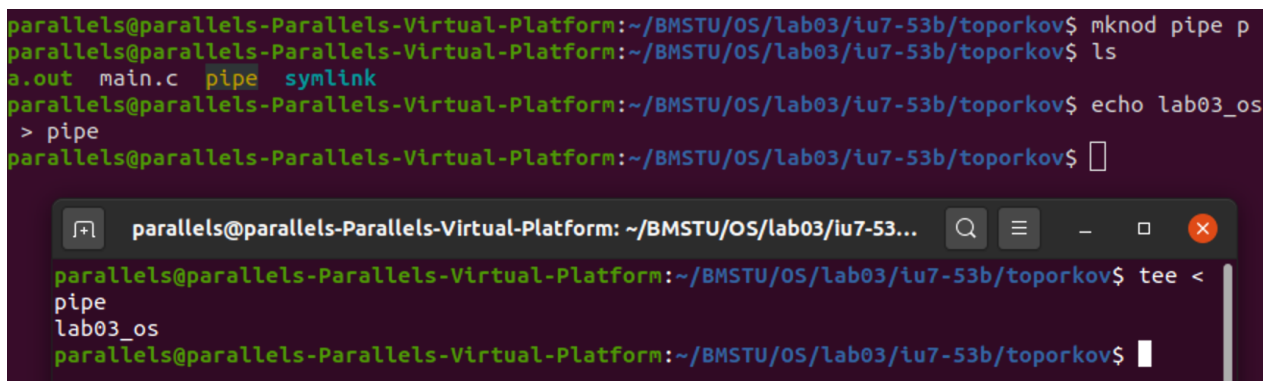
```
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$ pwd
/home/parallels/BMSTU/OS/lab03/iu7-53b/toporkov
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$ ln -s /home/p
arallels/BMSTU/OS/lab03/iu7-53b/toporkov/main.c /home/parallels/BMSTU/OS/lab03/iu7-53b/toporkov
/symlink
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$ ls -il
total 24
2232866 -rwxrwxr-x 1 parallels parallels 16816 Dec 14 16:02 a.out
2233489 -rw-rw-r-- 1 parallels parallels 347 Dec 14 16:01 main.c
2233504 lrwxrwxrwx 1 parallels parallels 54 Dec 14 16:43 symlink -> /home/parallels/BMSTU/OS
/lab03/iu7-53b/toporkov/main.c
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$
```

Рисунок 3.2: Создание символической ссылки.

4 Программные каналы

Демонстрация работы именованного программного канала:

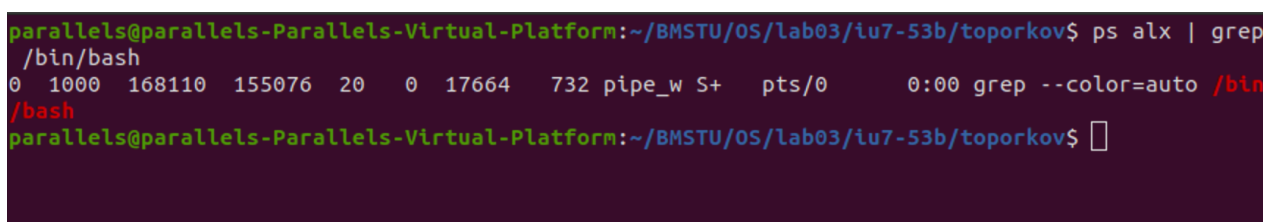
- создаем именованный программный канал командой `mknod` с именем `pipe`
- направляем текст в программный канал;
- меняем консоль;
- получаем через канал текст и, используя команду `tee`, выводим на экран;



```
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$ mknod pipe p
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$ ls
a.out main.c pipe symlink
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$ echo lab03_os
> pipe
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$

parallels@parallels-Parallels-Virtual-Platform: ~/BMSTU/OS/lab03/iu7-53b/toporkov$ tee <
pipe
lab03_os
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$
```

Рисунок 4.1: Демонстрация работы именованного программного канала.



```
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$ ps alx | grep
/bin/bash
0 1000 168110 155076 20 0 17664 732 pipe_w S+ pts/0 0:00 grep --color=auto /bin
/bash
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/iu7-53b/toporkov$
```

Рисунок 4.2: Демонстрация работы неименованного программного канала.

5 Приоритеты процессов

Создание процесса с приоритетом 20 с помощью команды `nice`.

```

parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/lu7-53b/toporkov
$ nice -n 20 ./a.out&
[1] 171173
parallels@parallels-Parallels-Virtual-Platform:~/BMSTU/OS/lab03/lu7-53b/toporkov
$
171173 171173 171173 171173 171173 171173 171173 171173 171173
3 171173 171173 171173 171173 171173 171173 171173 171173 171173
73 171173 171173 171173 171173 171173 171173 171173 171173 171173
173 171173 171173 171173 171173 171173 171173 171173 171173 171173
1173 171173 171173 171173 171173 171173 171173 171173 171173 171173
71173 171173 171173 171173 171173 171173 171173 171173 171173 171173
171173 171173 171173 171173 171173 171173 171173 171173 171173 171173
171173 171173 171173 171173 171173 171173 171173 171173 171173 171173
171173 171173 171173 171173 171173 171173 171173 171173 171173 171173
3 171173 171173 171173 171173 171173 171173 171173 171173 171173 171173
73 171173 171173 171173 171173 171173 171173 171173 171173 171173 171173
174 171173 171173 171173 171173 171173 171173 171173 171173 171173 171173
1174 171173 171173 171173 171173 171173 171173 171173 171173 171173 171173
71174 171173 171173 171173 171173 171173 171173 171173 171173 171173 171173
171174 171173 171173 171173 171173 171173 171173 171173 171173 171173 171173
171174 171173 171173 171173 171173 171173 171173 171173 171173 171173 171173
171174 171173 171173 171173 171173 171173 171173 171173 171173 171173 171173
4 171174 171174 171174 171174 171174 171174 171174 171174 171174 171174

```

Рисунок 5.1: Создание процесса с приоритетом 20 с помощью команды `nice`.