

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №4
за IV семестр
по дисциплине: "Операционные системы и системное программирование"
Тема: "Процессы"

Выполнил:
студент 2 курса
факультета ЭИС
группы ПО-4 (1)
Галанин П. И. (зачётка №190333)

Проверил:
ст. преподаватель
Давидюк Ю. И.

Лабораторная работа №4

Тема: "Процессы".

Цель:

Ход работы:

Задание для выполнения Написать программу, которая будет реализовывать следующие функции:

- сразу после запуска получает и сообщает свой ID и ID родительского процесса;
- перед каждым выводом сообщения об ID процесса и родительского процесса эта информация получается заново;
- порождает процессы, формируя генеалогическое дерево согласно варианту, сообщая, что "процесс с ID таким-то породил процесс с таким-то ID";
- перед завершением процесса сообщить, что "процесс с таким-то ID и таким-то ID родителя завершает работу";
- один из процессов должен вместо себя запустить программу, указанную в варианте задания.

На основании выходной информации программы предыдущего пункта изобразить генеалогическое дерево процессов (с указанием идентификаторов процессов). Объяснить каждое выведенное сообщение и их порядок в предыдущем пункте.

Варианты индивидуальных заданий В столбце fork описано генеалогическое древо процессов: каждая цифра указывает на относительный номер (не путать с pid) процесса, являющегося родителем для данного процесса. Например, строка 0 1 1 1 3 означает, что первый процесс не имеет родителя среди ваших процессов (порождается и запускается извне), второй, третий и четвертый - порождены первым, пятый - третьим. В столбце exes указан номер процесса, выполняющего вызов exes, команды для которого указаны в последнем столбце. Запускайте команду обязательно с какими-либо параметрами.

Таблица 1 – Вариант 5

No	fork	exec
5	0 1 1 2 2 3 3 5	df

					ЛР.ПО4.190333-04 81 00					
Изм	Лист	№ докум.	Подп.	Дата						
Разраб.	Галанин				Лабораторная работа №4 Процессы			Лит.	Лист	Листов
Пров.	Давидюк							Л	2	5
								БрГТУ		
Н. контр.										
Утв.										

Вариант 5

Листинг: main.c

```

1 #include <unistd.h>
2 #include <sys/types.h>
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 // 0112233
7 // 1234567
8
9 // if0
10 // if1
11 //     if2
12 //         if4
13 //             if5
14 //                 if3
15 //                     if6
16 //                         if7
17
18 int main()
19 {
20     pid_t pid;
21
22     // print 0
23     printf("start pr 0 PID = %d PPID = %d\n", getpid(), getppid());
24     //1
25     if ((pid = fork()) == -1) printf("Err\n");
26     else if (pid == 0)
27     {
28         //print 1
29         printf("\t\tstart pr 1 PID = %d PPID = %d\n", getpid(), getppid());
30         //2
31         if ((pid = fork()) == -1) printf("Err\n");
32         else if (pid == 0)
33         {
34             //print 2
35             printf("\t\t\tstart pr 2 PID = %d PPID = %d\n", getpid(), getppid());
36             //4
37             if ((pid = fork()) == -1) printf("Err\n");
38             else if (pid == 0)
39             {
40                 //print 4
41                 printf("\t\t\t\tstart pr 4 PID = %d PPID = %d\n", getpid(), getppid());
42                 printf("\t\t\t\tend pr 4\n");
43                 exit(0);
44             } else sleep(1);
45             //5
46             if ((pid = fork()) == -1) printf("Err\n");
47             else if (pid == 0)
48             {
49                 //print 5
50                 printf("\t\t\t\t\tstart pr 5 PID = %d PPID = %d\n", getpid(), getppid());
51                 printf("\t\t\t\t\tend and exec(df) pr 5\n");
52                 printf("\n");
53                 printf("\t\t\t\t\t== run $df\n");
54                 execl("/bin/df", "df", NULL);
55             } else sleep(1);
56         }
57     }
58     exit(0);
59 }
```

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96

```
else sleep(3);
//3
if ((pid = fork()) == -1) printf("Err\n");
else if (pid == 0)
{
    //print 3
    printf("\t\t\tstart pr 3 PID = %d PPID = %d\n", getpid(), getppid());
    //6
    if ((pid = fork()) == -1) printf("Err\n");
    else if (pid == 0)
    {
        //print 6
        printf("\t\t\t\t\tstart pr 6 PID = %d PPID = %d\n", getpid(), getppid());
        printf("\t\t\t\t\tend pr 6\n");
        exit(0);
    } else sleep(1);

    //7
    if ((pid = fork()) == -1) printf("Err\n");
    else if (pid == 0)
    {
        //print 7
        printf("\t\t\t\t\t\t\tstart pr 7 PID = %d PPID = %d\n", getpid(), getppid());
        printf("\t\t\t\t\t\t\tend pr 7\n");
        exit(0);
    } else sleep(1);

    printf("\t\t\tend pr 3\n");
    exit(0);
}
else sleep(3);
printf("\t\tend pr 1\n");
exit(0);
}
else sleep(10);

printf("End program...\n");
return 0;
}
```

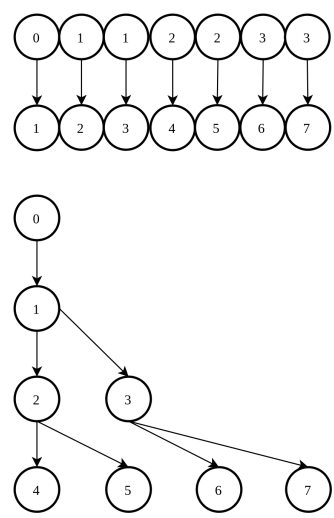


Рисунок 1 – Генелогическое дерево варианта 5

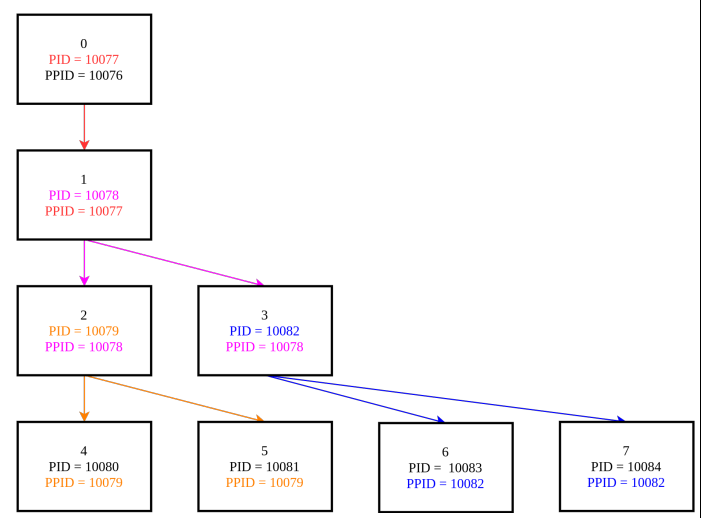


Рисунок 2 – Проверка генелогического дерева

Листинг: Out

```
1 start pr 0 PID = 8661 PPID = 8660
2     start pr 1 PID = 8662 PPID = 8661
3         start pr 2 PID = 8663 PPID = 8662
4             start pr 4 PID = 8664 PPID = 8663
5             end pr 4
6             start pr 5 PID = 8665 PPID = 8663
7         end and exec(df) pr 5
8
9         == == == run $df
10 Filesystem      1K- blocks      Used Available Use% Mounted on
11 udev             1905716          0   1905716   0% /dev
12 tmpfs            387056         1636    385420   1% /run
13 /dev/sda5        28705700 17692176   9532308  65% /
14 tmpfs            1935264        20108   1915156   2% /dev/shm
15 tmpfs             5120           4      5116   1% /run/lock
16 tmpfs            1935264          0   1935264   0% /sys/fs/cgroup
17 /dev/loop0        56832         56832          0 100% /snap/core18/1988
18 /dev/loop1       134784        134784          0 100% /snap/docker/796
19 /dev/loop2        33152         33152          0 100% /snap/snapd/11107
20 /dev/loop3        33152         33152          0 100% /snap/snapd/11402
21 /dev/sda1         98304         36511     61793  38% /boot/efi
22 /dev/sda6       165168784 65692132   91016828  42% /home
23 tmpfs             387052          44    387008   1% /run/user/1000
24     start pr 3 PID = 8666 PPID = 8662
25         start pr 6 PID = 8667 PPID = 8666
26         end pr 6
27         start pr 7 PID = 8668 PPID = 8666
28         end pr 7
29     end pr 3
30 end pr 1
31 End program ...
```