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

Лабораторная работа №4
По дисциплине: «Операционные системы и системное программирование»
Тема: «Процессы»

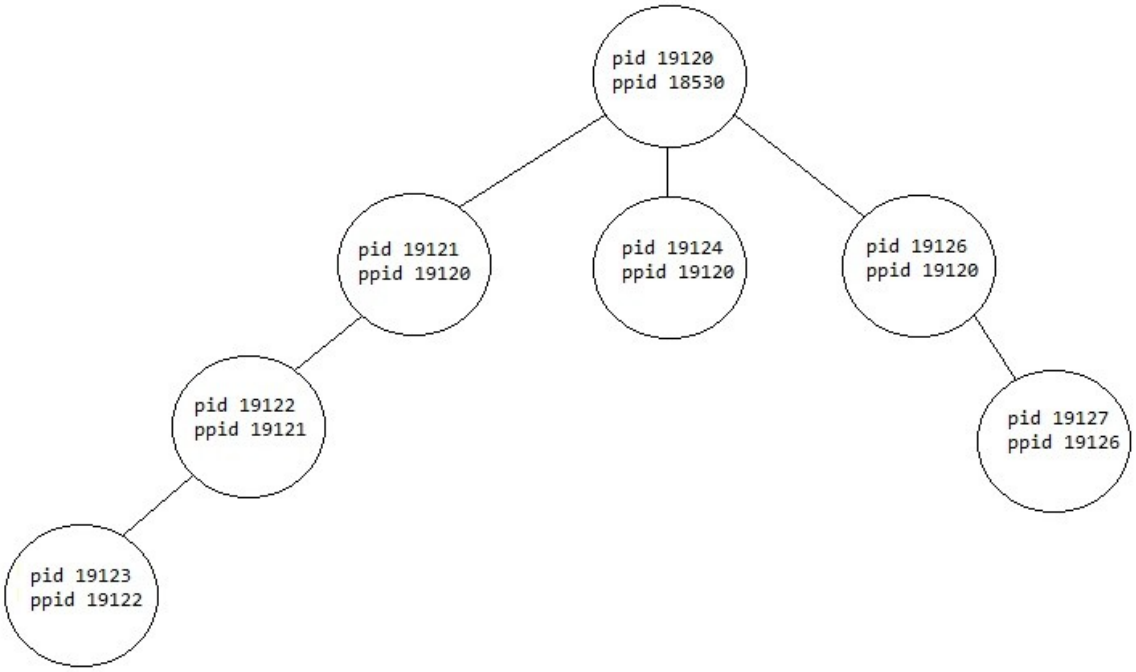
Подготовил:
Студент 2 курса
Группы ПО-3(2)
Огиевич Е.А.
Проверила:
Давидюк Ю.И.

Брест, 2020

Лабораторная работа №4

Цель: изучить работу с процессами и компилятором GCC.

Задание:
• 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.



Код программы(https://hastebin.com/boqisofugo.pl):

```
#include <stdlib.h>
int main()
{
	printf("Процесс 1:\n\tPID - %d,\n\tPPID - %d\n", getpid(), getppid());

	int pid;

	if((pid = fork()) == -1)
	{
		printf("Ошибка: неудалось создать процесс.\n");
	}
```

```
	else if(pid == 0)
	{
		printf("Порождение процесса 2:\n\tPID - %d,\n\tPPID - %d\n", getpid(), getppid());

		if((pid = fork()) == -1)
		{
			printf("Ошибка: неудалось создать процесс.\n");
		}

		else if(pid == 0)
		{
			printf("Порождение процесса 5:\n\tPID - %d,\n\tPPID - %d\n", getpid(),
getppid());
			if((pid = fork()) == -1)
			{
				printf("Ошибка: неудалось создать процесс.\n");
			}
			else if(pid == 0)
			{
				printf("Порождение процесса 6:\n\tPID - %d,\n\tPPID - %d\n",
getpid(), getppid());

				printf("Завершение процесса 6:\n\tPID - %d,\n\tPPID - %d\n",
getpid(), getppid());
				execl("/usr/bin/free","free", NULL);
				exit(0);

			} else sleep(1);
			printf("Завершение процесса 5:\n\tPID - %d,\n\tPPID - %d\n", getpid(),
getppid());
			exit(0);
		} else sleep(2);
		printf("Завершение процесса 2:\n\tPID - %d,\n\tPPID - %d\n", getpid(), getppid());
		exit(0);
	} else sleep(1);


	sleep(1);
	if((pid = fork()) == -1)
	{
		printf("Ошибка: неудалось создать процесс.\n");
	}

	else if(pid == 0)
	{
		printf("Порождение процесса 3:\n\tPID - %d,\n\tPPID - %d\n", getpid(), getppid());
		printf("Завершение процесса 3:\n\tPID - %d,\n\tPPID - %d\n", getpid(), getppid());
		exit(0);
	} else sleep(1);

	if((pid = fork()) == -1)
	{
```

```
		printf("Ошибка: неудалось создать процесс.\n");
	}

	else if(pid == 0)
	{
		printf("Порождение процесса 4:\n\tPID - %d,\n\tPPID - %d\n", getpid(), getppid());

		if((pid = fork()) == -1)
		{
			printf("Ошибка: неудалось создать процесс.\n");
		}
		else if(pid == 0)
		{
			printf("Порождение процесса 7:\n\tPID - %d,\n\tPPID - %d\n", getpid(),
getppid());
			printf("Завершение процесса 7:\n\tPID - %d,\n\tPPID - %d\n", getpid(),
getppid());
			exit(0);
		} else sleep(1);
		printf("Завершение процесса 4:\n\tPID - %d,\n\tPPID - %d\n", getpid(), getppid());
		exit(0);
	} else sleep(1);

	sleep(1);

	printf("Завершение процесса 1:\n\tPID - %d,\n\tPPID - %d\n", getpid(), getppid());

	exit(0);

	return 0;
}
```

Процессы создаются и завершаются в логическом порядке, как указано по дереву. PPID следующего процесса соответствует PID родительского.

```
natefoust@natefoust-X550LC:~/OS/hello$ gcc hello.c
natefoust@natefoust-X550LC:~/OS/hello$ ./a.out
Процесс 1:
        PID - 19120,
        PPID - 18530
Порождение процесса 2:
        PID - 19121,
        PPID - 19120
Порождение процесса 5:
        PID - 19122,
        PPID - 19121
Порождение процесса 6:
        PID - 19123,
        PPID - 19122
Завершение процесса 6:
        PID - 19123,
        PPID - 19122
              всего        занято        свободно      общая  буф./врем.   доступно
Память:     8034008     2163228     3321124      674560     2549656     4891520
Подкачка:     2097148           0     2097148
Завершение процесса 5:
        PID - 19122,
        PPID - 19121
Завершение процесса 2:
        PID - 19121,
        PPID - 19120
Порождение процесса 3:
        PID - 19124,
        PPID - 19120
Завершение процесса 3:
        PID - 19124,
        PPID - 19120
Порождение процесса 4:
        PID - 19126,
        PPID - 19120
Порождение процесса 7:
        PID - 19127,
        PPID - 19126
Завершение процесса 7:
        PID - 19127,
        PPID - 19126
Завершение процесса 4:
        PID - 19126,
        PPID - 19120
Завершение процесса 1:
        PID - 19120,
        PPID - 18530
natefoust@natefoust-X550LC:~/OS/hello$
```

Вывод: ознакомился с компилятором GCC, вспомнил язык C, научился работать с процессами.