Лабораторная 3

Цель: знакомство с процессами Linux.
Упражнение 1. Протестируйте программы, рассмотренные на Лекции 3.
Упражнение 2. Создайте процесс с помощью вызова fork, с помощью команд ps
и grep получите информацию о созданных вами родительском и дочернем процессах. Используя команду kill убейте родительский процесс, продолжил ли выполняться дочерний процесс?
Упражнение 3. Создайте дерево процессов с помощью вызова fork. С помощью команды pstree найдите поддерево созданных процессов. В каталоге /proc виртуальной файловой системы найдите папки с именами, совпадающими с идентификаторами созданных процессов, и просмотрите содержимое папок task/children.

Упражнение 1. Протестируйте программы, рассмотренные на Лекции 3.

```c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
void oldman();
void recreation();
int main() {
  pid_t child_pid, parent_pid;
  int i = 0;
  fprintf(stdout, "Before RECREATION %i\n", parent_pid =
(int)getpid());
  child_pid = fork();
  while (i++ < 5)
    if (child_pid != 0)
      oldman();
    else
      recreation();
  return 0;
}
void oldman() {
  fprintf(stdout, "I'm not yet dead! My ID is %i\n", (int)getpid());
}
void recreation() { fprintf(stdout, "Who I am? My ID is %i\n",
(int)getpid()); }
```

result:

```
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ gcc -o lab3 lab3.c
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ ./lab3
Before RECREATION 257532
I'm not yet dead! My ID is 257532
I'm not yet dead! My ID is 257532
I'm not yet dead! My ID is 257532
I'm not yet dead! My ID is 257532
I'm not yet dead! My ID is 257532
Who I am? My ID is 257533
Who I am? My ID is 257533
Who I am? My ID is 257533
Who I am? My ID is 257533
Who I am? My ID is 257533
```

```c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main() {
  pid_t child_pid, parent_pid;
  double s = 0.0;
  child_pid = fork();
  if (child_pid != 0) {
    s += 3.14;
    fprintf(stdout, "CHILD: %i s=%g &s=%p\n", (int)getpid(), s,
(void*)&s);
  } else {
    s += 2.72;
    fprintf(stdout, "PARENT: %i s=%g &s=%p\n", (int)getpid(), s,
(void*)&s);
  }
  return 0;
}
```

result:

```
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ gcc -o lab3_1 lab3_1.c
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ ./lab3_1
CHILD: 264319 s=3.14 &s=0x7ffd3da713d0
PARENT: 264320 s=2.72 &s=0x7ffd3da713d0
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$
```

```c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main() {
  pid_t child_pid;
  pid_t parent_pid;
  double s = 0.0;
  FILE* fp;

  child_pid = fork();
  fp = fopen("test.dat", "a+");

  if (child_pid != 0) {
    s += 3.14;
```

```
    fprintf(fp, "CHILD: %i s=%g &s=%p fp=%p\n", (int)getpid(), s,
(void*)&s,
            (void*)fp);
  } else {
    s += 2.72;
    fprintf(fp, "PARENT: %i s=%g &s=%p fp=%p\n", (int)getpid(), s,
(void*)&s,
            (void*)fp);
  }

  fclose(fp);
  return 0;
}
```

```
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ ./lab3_2
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ cat test.dat
CHILD: 287729 s=3.14 &s=0x7ffe1c289d18 fp=0x564f26a522a0
PARENT: 287730 s=2.72 &s=0x7ffe1c289d18 fp=0x564f26a522a0
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$
```

Упражнение 2.

```
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ gcc -o 3 lab3.c
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ ./3
Before RECREATION 7120
I'm not yet dead! My ID is 7120
I'm not yet dead! My ID is 7120
I'm not yet dead! My ID is 7120
I'm not yet dead! My ID is 7120
I'm not yet dead! My ID is 7120
Who I am? My ID is 7121
Who I am? My ID is 7121
Who I am? My ID is 7121
Who I am? My ID is 7121
Who I am? My ID is 7121
```

Получение информации о процессах:

```
tania@TaniaLaptop:~$ ps 7120
    PID TTY      STAT   TIME COMMAND
   7120 pts/0    S+     0:00 ./3
tania@TaniaLaptop:~$ ps 7121
    PID TTY      STAT   TIME COMMAND
   7121 pts/0    S+     0:00 ./3
tania@TaniaLaptop:~$ ps aux |grep 7121
tania       7121  0.0  0.0   2776   100 pts/0    S+   15:42   0:00 ./3
tania       7751  0.0  0.0   4028  2048 pts/5    S+   15:44   0:00 grep --color=auto 7121
tania@TaniaLaptop:~$ ps aux |grep 7120
tania       7120  0.0  0.0   2776   980 pts/0    S+   15:42   0:00 ./3
tania       7821  0.0  0.0   4028  1992 pts/5    S+   15:44   0:00 grep --color=auto 7120
tania@TaniaLaptop:~$
```

завершить родительский процесс, используя команду kill с PID родительского процесса

```
tania@TaniaLaptop:~$ ps -a
    PID TTY          TIME CMD
    672 pts/1    00:00:00 bash
   3453 pts/2    00:00:00 sh
   3459 pts/2    00:00:00 sh
   3463 pts/2    00:00:02 node
   3505 pts/2    00:00:00 node
   3525 pts/2    00:00:19 node
   3683 pts/2    00:00:02 cpptools
   3741 pts/2    00:00:00 cpptools-srv
   4264 pts/2    00:00:00 cpptools-srv
   4289 pts/2    00:00:00 cpptools-srv
   5000 pts/2    00:00:00 cpptools-srv
   7120 pts/0    00:00:00 3
   7121 pts/0    00:00:00 3
   8599 pts/5    00:00:00 ps
tania@TaniaLaptop:~$ kill 7120
tania@TaniaLaptop:~$ ps -a
    PID TTY          TIME CMD
    672 pts/1    00:00:00 bash
   3453 pts/2    00:00:00 sh
   3459 pts/2    00:00:00 sh
   3463 pts/2    00:00:03 node
   3505 pts/2    00:00:00 node
   3525 pts/2    00:00:20 node
   3683 pts/2    00:00:02 cpptools
   3741 pts/2    00:00:00 cpptools-srv
   4264 pts/2    00:00:00 cpptools-srv
   4289 pts/2    00:00:00 cpptools-srv
   5000 pts/2    00:00:00 cpptools-srv
   7121 pts/0    00:00:00 3
   9101 pts/5    00:00:00 ps
tania@TaniaLaptop:~$
```

Упражнение 3. Создание С-программы, которая создает дерево процессов

```c
#include <signal.h>
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
void createNewProcess() {
  pid_t child_pid = fork();
  if (child_pid != 0) {
    fprintf(stdout, "%u create process %u\n", getpid(), child_pid);
  } else {
    return;
  }
}
int main() {
  pid_t child_pid, parent_pid;
  int i = 0;
  fprintf(stdout, "First process %i\n", parent_pid = (int)getpid());
  child_pid = fork();
  createNewProcess();
  createNewProcess();
  getchar();
  return 0;
}
```

```
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ gcc -o 3 lab3_3.c
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ ./3
Before RECREATION 18055
18055 spawned 18057
18056 spawned 18058
18056 spawned 18061
18055 spawned 18059
18057 spawned 18060
18058 spawned 18062
```

команда pstree, чтобы увидеть дерево процессов:

```
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ ps -a
    PID TTY          TIME CMD
    672 pts/1    00:00:00 bash
   3453 pts/2    00:00:00 sh
   3459 pts/2    00:00:00 sh
   3463 pts/2    00:00:04 node
   3505 pts/2    00:00:00 node
   3525 pts/2    00:00:52 node
   3683 pts/2    00:00:04 cpptools
   3741 pts/2    00:00:00 cpptools-srv
   4264 pts/2    00:00:00 cpptools-srv
   4289 pts/2    00:00:00 cpptools-srv
   5000 pts/2    00:00:00 cpptools-srv
  18055 pts/6    00:00:00 3
  18056 pts/6    00:00:00 3
  18057 pts/6    00:00:00 3
  18058 pts/6    00:00:00 3
  18059 pts/6    00:00:00 3
  18060 pts/6    00:00:00 3
  18061 pts/6    00:00:00 3
  18062 pts/6    00:00:00 3
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ pstree
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ pstree 18055
3─┬─3─┬─3───3
  │   └─3
  ├─3───3
  └─3
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ pstree 18056
3─┬─3───3
  └─3
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$ pstree 18057
3───3
tania@TaniaLaptop:~/5sem/os/os_labs/lab3$
```

содержимое файла children для pid 18055

```
tania@TaniaLaptop:/proc/18055/task$ cd 18055/
tania@TaniaLaptop:/proc/18055/task/18055$ ls
arch_status  cgroup      cmdline  cwd      fd       io        maps       mounts  oom_adj        pagemap      root       sessionid  smaps_rollup  statm    uid_map
attr         children    comm     environ  fdinfo   limits    mem        net     oom_score      personality  sched      setgroups  stack         status   wchan
auxv         clear_refs  cpuset   exe      gid_map  loginuid  mountinfo  ns      oom_score_adj  projid_map   schedstat  smaps      stat          syscall
tania@TaniaLaptop:/proc/18055/task/18055$ children
children: command not found
tania@TaniaLaptop:/proc/18055/task/18055$ cat children
18056 18057 18059 tania@TaniaLaptop:/proc/18055/task/18055$
```

```
tania@TaniaLaptop:/proc/18055/task/18055$ cat children
18056 18057 18059 tania@TaniaLaptop:/proc/18055/task/18055$
```