

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”  
КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ УПРАВЛІННЯ

Звіт з лабораторної роботи № 3  
з дисципліни “Основи архітектури ЕОМ та операційні системи”

Виконав: ст. гр. КН-221В

Шулюпов Єгор Русланович

Перевірив: Асистент каф. ПІІТУ

Дмитро Миколайович Ковальчук

Харків

2022

Передача параметрів командного рядка до програми.  
Робота із процесами. Одержання системної інформації.

### ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

Розробити дві програми для Linux, перша з яких:

- а) при заданні ключа -w ім'я-виконуваного файлу
  - запускає заданий додаток (з пошуком його у системному шляху, заданому змінною оточення PATH)
  - очікує завершення програми та виводить повідомлення про завершення на екран разом із кодом повернення
- б) при заданні ключа -f ім'я-виконуваного файлу - запускає заданий додаток (з пошуком його у системному шляху, заданому змінною оточення PATH) - негайно продовжує виконання (асинхронний запуск)

Друга програма повинна виводити на екран:

- Значення аргументів командного рядка;
- ідентифікатори: процесу (pid), сесії (sid), групи процесів (pgid) та процесу-предка (ppid)
- ім'я користувача, під яким процес запущено та його групу (отримані з файлів /etc/passwd та /etc/group)

## Програма : №1

home > vorpolochek > lab\_EOM > #3 > Lab3 > C LAB3.c

```
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <sys/types.h>
4  #include <stdlib.h>
5  #include <sys/wait.h>
6  #include <string.h>
7  int main(int argc, char* argv[])
8  {
9      int c = 0;
10     pid_t pid;
11     while((c = getopt(argc, argv, "w:f:")) != -1){
12         switch (c){
13             case 'w':
14                 printf ("Использование [-w] %s\n", optarg);
15                 if ((pid = fork ()) < 0){
16                     exit (-1);
17                 }
18                 if(pid == 0){
19                     if (execvp ("./LAB_3", argv) == -1) {
20                         printf ("Ошибка запуска программы!");
21                         exit(-1);}
22                     else {
23                         printf ("Потомок начался с кодом %d\n", pid);
24                         int status;
```

```
25         waitpid(pid, &status, 0);
26         if(WIFEXITED (status)){
27             printf("Потомок закончился с кодом %d\n", WEXITSTATUS(status));} }
28         break;
29             case 'f':
30                 printf ("Использование [-f] %s \n", optarg);
31                 if ((pid = fork ()) < 0){
32                     exit (-1);}
33                 if(pid == 0){
34                     if (execvp ("./LAB_3", argv) == -1) {
35                         printf ("Ошибка запуска программы!");
36                         exit(-1);}
37                     else
38                     { sleep (10);
39                     exit(0);} break;
40             default:
41                 printf("Ошибка!\n");
42                 break;
43             }return 0;
44     }
```

home > vorpolochek > lab\_EOM > #3 > Lab3 > C LAB\_3.c

```
1  #include<unistd.h>
2  #include<stdio.h>
3
4  int main(int arg, char* argv[])
5  {
6
7      printf("Программа выполняется!\n");
8      printf("Программа закончила выполнение!\n");
9
10     return 0;
11 }
```

```
vorpolochek@vorpolochek-VirtualBox:~/lab_ECM/#3/Lab3$ ./LAB3 -w LAB3
Использование [-w] LAB3
Потсмок начался с кодом 6907
Программа выполняется!
Программа закончила выполнение!
Потсмок закончился с кодом 0
vorpolochek@vorpolochek-VirtualBox:~/lab_ECM/#3/Lab3$ ./LAB3 -f LAB3
Использование [-f] LAB3
Программа выполняется!
Программа закончила выполнение!
```

## Програма : №2

```
vorpolochek@vorpolochek-VirtualBox:~/lab_EOM/#3$ ./main2
Argument:      (null)
PID:           7268
SID:           4473
PGID:          7268
PPID:          4473
Username:      vorpolochek
Group name:    vorpolochek
```

```
home > vorpolochek > lab_EOM > #3 > C main2.c
1  #include <stdio.h>
2  #include <getopt.h>
3  #include <unistd.h>
4  #include <sys/types.h>
5  #include <pwd.h>
6  #include <stdlib.h>
7  #include <grp.h>
8  #include <sys/wait.h>
9  #include <errno.h>
10 int main(int argc, char** argv)
11 {
12     struct passwd *pwd;
13     pwd = getpwuid(getuid());
14     struct group *grp;
15     grp = getgrgid(getgid());
16     pid_t pid;
17     printf("Argument:\t%s\n", argv[1]);
18     printf("PID:\t%d\n", getpid());
19     printf("SID:\t%d\n", getsid(0));
20     printf("PGID:\t%d\n", getpgid(0));
21     printf("PPID:\t%d\n", getppid());
22     printf("Username:\t%s\n", pwd->pw_name);
23     printf("Group name:\t%s\n", grp->gr_name);
24     return 0;
25 }
```

## Висновок

Виконавши лабораторну роботу №3, я ознайомився з передачею параметрів командного рядка до програми, обробкою параметрів командного рядка Linux, перенаправленням введення-виводу та каналами та здійснювати роботу з процесорами.