

Санкт-Петербургский Государственный Политехнический Университет
Кафедра «Прикладная математика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

По курсу “Операционные системы”

Выполнил студент гр. 4057/2:

Зенцев Ф.К.

Преподаватель:

Тимофеев Д.А.

Санкт-Петербург 2010

Формулировка задания (3.4): Напишите программу-демон, получающую в качестве аргумента число N и через N минут после запуска отправляющую пользователю, от чьего имени она запущена, письмо с произвольным текстом (запустите программу mail).

Реализация: На языке C. Для написания программы-демона требуется создать собственный процесс и удалить процесс-родителя, тогда созданный процесс подхватится процессом INIT - такие процессы и называют демонами. Также для безопасности стоит закрыть стандартные потоки ввода/вывода. Все равно программа-демон не взаимодействует с терминалом, а оставшиеся открытыми они могут представлять потенциальную опасность. Далее по заданию выжидается время с помощью функции SLEEP. Сообщение отправляется с помощью вызова функции SYSTEM - вызывается программа MAIL. Имя пользователя, запустившего программу, можно узнать с помощью функций GETUID, GETPWUID.

Исходный код программы

Listing 1: Simple daemon program example source

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #include <syslog.h>
5 #include <unistd.h>
6 #include <sys/types.h>
7 #include <sys/stat.h>
8 #include <pwd.h>
9
10 int prepare_daemon( void )
11 {
12     pid_t pid, sid;
13
14     pid = fork();
15
16     if (pid < 0)
17     {
18         return -1;
19     }
20     else if (pid > 0)
21     {
22         exit(EXIT_SUCCESS);
23     }
24
25     umask(0);
26
27     /* It is possible to open log file here */
28
29     sid = setsid();
30
31     if (sid < 0)
32     {
33         exit(EXIT_FAILURE);
34     }
35
36     if (chdir("/") < 0)
37     {
38         exit(EXIT_FAILURE);
39     }
40
41     close(STDIN_FILENO);
42     close(STDOUT_FILENO);
43     close(STDERR_FILENO);
44 }
45
46 void print_help( void )
47 {
48     printf("Usage: _lab3_N\n");
49 }
50
51 int main( int argc, char *argv[] )
```

```

52 {
53     int N;
54     struct passwd *pwd = getpwuid(getuid());
55     char *user_name = pwd->pw_name;
56
57     if (argc < 2)
58     {
59         print_help();
60         return EXIT_FAILURE;
61     }
62
63     N = atoi(argv[1]);
64
65     if (N == 0)
66     {
67         printf("Incorrect input number\n");
68         return EXIT_FAILURE;
69     }
70
71     printf("Everything is OK. Going to start daemon for %i seconds.\n", N * 60);
72
73     if (prepare_daemon() == -1)
74     {
75         return EXIT_FAILURE;
76     }
77
78     sleep(N * 60);
79
80     {
81         char Buf[300];
82
83         sprintf(Buf, "echo \\\"%s\\n\\nLet me introduce myself. I am daemon.\\\" | mail -s \\\"
            Greetings.\\\" ", user_name);
84         system(Buf);
85     }
86
87     exit(EXIT_SUCCESS);
88
89     return EXIT_SUCCESS;
90 }

```