Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет

к лабораторной работе №1 по курсу
«Операционные системы»

Студент: Батбилэг Н.

Группа: ИУ7-71Б

Оценка (баллы) _____

Преподаватель: Рязанова Н.Ю.

Москва.
2021 г.

1. Задание

Написать программу, которая создавала бы процесс-демон, а также выводить время.

2. Реализация задания

В листингах 1-2 представлен код задания. 494, 434, 397, 504

Листинг 1. prog.c

```
#include "apue.h"
#include <syslog.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/resource.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdio.h>
#include <signal.h>
#include <string.h>
#include <errno.h>
#include <sys/file.h>
#include <pthread.h>

sigset_t mask;

#define LOCKFILE "/var/run/daemon.pid" //нужны права суперпользователя, чтобы
                                       //создавать файлы в этой директории
#define LOCKMODE (S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH)

int lockfile(int fd)
{
    struct flock fl;
    fl.l_type = F_WRLCK;
    fl.l_start = 0;
    fl.l_whence = SEEK_SET;
    fl.l_len = 0;
    return(fcntl(fd, F_SETLK, &fl));
}
int already_running(void)
{
    syslog(LOG_ERR, "Проверка на многократный запуск!");

    int fd;
    char buf[16];

    fd = open(LOCKFILE, O_RDWR | O_CREAT, LOCKMODE);

    if (fd < 0)
    {
        syslog(LOG_ERR, "невозможно открыть %s: %s!", LOCKFILE, strerror(errno));
        exit(1);
    }

    syslog(LOG_WARNING, "Lock-файл открыт!");

    lockfile(fd);
    if (errno == EWOULDBLOCK) {
        syslog(LOG_ERR, "невозможно установить блокировку на %s: %s!", LOCKFILE,
                                                                    strerror(errno));
        close(fd);
        exit(1);
```

```
53      }
54
55      syslog(LOG_WARNING, "Записываем PID!");
56
57      ftruncate(fd, 0);
58      sprintf(buf, "%ld", (long)getpid());
59      write(fd, buf, strlen(buf) + 1);
60
61      syslog(LOG_WARNING, "Записали PID!");
62
63      return 0;
64  }
65
66  void daemonize(const char *cmd)
67  {
68      int fd0, fd1, fd2;
69      pid_t pid;
70      struct rlimit rl;
71      struct sigaction sa;
72
73      // 1. Сбрасывание маски режима создания файла
74      umask(0);
75
76      // 2. Получение максимального возможного номера дискриптора
77      if (getrlimit(RLIMIT_NOFILE, &rl) < 0)
78          perror("Невозможно получить максимальный номер дискриптора!\n");
79
80      // 3. Стать лидером новой сессии, чтобы утратить управляющий терминал
81      if ((pid = fork()) < 0)
82          perror("Ошибка функции fork!\n");
83      else if (pid != 0) //родительский процесс
84          exit(0);
85
86      setsid();
87
88      // 4. Обеспечение невозможности обретения терминала в будущем
89      sa.sa_handler = SIG_IGN;
90      sigemptyset(&sa.sa_mask);
91      sa.sa_flags = 0;
92      if (sigaction(SIGHUP, &sa, NULL) < 0)
93          perror("Невозможно игнорировать сигнал SIGHUP!\n");
94      if ((pid = fork()) < 0)
95          perror("Ошибка функции fork!\n");
96      else if (pid != 0) //родительский процесс
97          exit(0);
98
99      // 5. Назначить корневой каталог текущим рабочим каталогом,
100     // чтобы впоследствии можно было отмонтировать файловую систему
101     if (chdir("/") < 0)
102         perror("Невозможно назначить корневой каталог текущим рабочим каталогом!\n");
103
104     // 6. Закрыть все файловые дескрипторы
105     if (rl.rlim_max == RLIM_INFINITY)
106         rl.rlim_max = 1024;
107     for (int i = 0; i < rl.rlim_max; i++)
108         close(i);
109
110     // 7. Присоеденить файловые дескрипторы 0, 1, 2 к /dev/null
111     fd0 = open("/dev/null", O_RDWR);
112     fd1 = dup(0); //копируем файловый дискриптор
```

```
113     fd2 = dup(0);
114
115     // 8. Инициализировать файл журнала
116     openlog(cmd, LOG_CONS, LOG_DAEMON);
117     if (fd0 != 0 || fd1 != 1 || fd2 != 2)
118     {
119         syslog(LOG_ERR, "ошибочные файловые дескрипторы %d %d %d\n", fd0, fd1, fd2);
120         exit(1);
121     }
122
123     syslog(LOG_WARNING, "Демон запущен!");
124
125 }
126
127 void reread(void)
128 {
129     /* ... */
130 }
131
132 void *thr_fn(void *arg)
133 {
134     int err, signo;
135
136     for(;;) {
137         err = sigwait(&mask, &signo);
138         if (err != 0) {
139             syslog(LOG_ERR, "ошибка вызова функции sigwait");
140             exit(1);
141         }
142         switch (signo) {
143             case SIGHUP:
144                 syslog(LOG_INFO, "Чтение конфигурационного файла");
145                 reread();
146                 break;
147             case SIGTERM:
148                 syslog(LOG_INFO, "получен сигнал SIGTERN; выход");
149                 exit(0);
150             default:
151                 syslog(LOG_INFO, "получен непредвиденный сигнал %d\n", signo);
152         }
153     }
154     return(0);
155 }
156
157 int main(int argc, char *argv[])
158 {
159     int err;
160     pthread_t tid;
161     daemonize("lab1");
162
163     // Блокировка файла для одной существующей копии демона
164     if (already_running() != 0)
165     {
166         syslog(LOG_ERR, "Демон уже запущен!\n");
167         exit(1);
168     }
169
170     // Создать поток, который будет заниматься обработкой SIGHUP и SIGTERM
171     err = pthread_create(&tid, NULL, thr_fn, 0);
172     if (err != 0)
```

```
            perror("невозможно создать поток");
173
174     syslog(LOG_WARNING, "Проверка пройдена!");
175
176     while(1)
177     {
178         syslog(LOG_INFO, "Демон!");
179         sleep(5);
180     }
181
182     exit(0);
    }
```

На рисунках 1-2 представлен список запущенных процессов с моим демоном с ID 13081, где:

PPID – ID родительского процесса

PID – ID процесса

PGID – ID группы процессов

SID – ID сессии

TTY – имя терминала

TPGID – ID группы процесса

STAT – состояние процесса

UID – ID пользователя

TIME – время использования процессора для конкретного процесса

COMMAND – строка команды

```
dalai@dalai-Inspiron-5567:~/Desktop/BMSTU/labs/lab_01$ sudo ./a.out
[sudo] password for dalai:
dalai@dalai-Inspiron-5567:~/Desktop/BMSTU/labs/lab_01$ ps -ajx
   PPID     PID    PGID     SID TTY        TPGID STAT   UID   TIME COMMAND
      0       1       1       1 ?           -1 Ss       0   0:03 /sbin/init sp
      0       2       0       0 ?           -1 S        0   0:00 [kthreadd]
      2       3       0       0 ?           -1 I<       0   0:00 [rcu_gp]
      2       4       0       0 ?           -1 I<       0   0:00 [rcu_par_gp]
      2       6       0       0 ?           -1 I<       0   0:00 [kworker/0:0H
      2       9       0       0 ?           -1 I<       0   0:00 [mm_percpu_wq
      2      10       0       0 ?           -1 S        0   0:00 [rcu_tasks_ru
      2      11       0       0 ?           -1 S        0   0:00 [rcu_tasks_tr
      2      12       0       0 ?           -1 S        0   0:00 [ksoftirqd/0]
      2      13       0       0 ?           -1 I        0   0:10 [rcu_sched]
      2      14       0       0 ?           -1 S        0   0:00 [migration/0]
      2      15       0       0 ?           -1 S        0   0:00 [idle_inject/
      2      16       0       0 ?           -1 S        0   0:00 [cpuhp/0]
      2      17       0       0 ?           -1 S        0   0:00 [cpuhp/1]
      2      18       0       0 ?           -1 S        0   0:00 [idle_inject/
      2      19       0       0 ?           -1 S        0   0:00 [migration/1]
      2      20       0       0 ?           -1 S        0   0:00 [ksoftirqd/1]
      2      22       0       0 ?           -1 I<       0   0:00 [kworker/1:0H
```

Рисунок 1. Список запущенных процессов (начало).

```
  11566   11613   11564   11564 ?            -1 Sl    1000   0:41 /snap/code/82
  11564   11623   11564   11564 ?            -1 Sl    1000   3:22 /snap/code/82
  11623   11642   11564   11564 ?            -1 Sl    1000   0:14 /snap/code/82
  11564   11654   11564   11564 ?            -1 Sl    1000   0:10 /snap/code/82
  11654   11672   11564   11564 ?            -1 Sl    1000   0:03 /snap/code/82
  11672   11725   11725   11725 pts/2    11725 Ss+   1000   0:00 /usr/bin/bash
  11642   11920   11564   11564 ?            -1 Sl    1000   0:14 /home/dalai/.
  11920   11949   11564   11564 ?            -1 Sl    1000   0:04 /home/dalai/.
      2   12436       0       0 ?            -1 I        0   0:00 [kworker/u8:2
      2   12475       0       0 ?            -1 D<       0   0:00 [kworker/u9:1
      2   12641       0       0 ?            -1 I        0   0:00 [kworker/1:2-
      2   12727       0       0 ?            -1 I        0   0:00 [kworker/2:1-
      2   12741       0       0 ?            -1 I        0   0:00 [kworker/0:0-
      2   12892       0       0 ?            -1 I        0   0:00 [kworker/u8:3
   5664   12957    1847    1847 ?            -1 Sl    1000   0:00 /opt/google/c
   1588   13081   13081   13081 ?            -1 Ss       0   0:00 ./a.out
      2   13102       0       0 ?            -1 I<       0   0:00 [kworker/u9:2
  11300   13104   13104   11300 pts/0    13104 R+    1000   0:00 ps -ajx
dalai@dalai-Inspiron-5567:~/Desktop/BMSTU/labs/lab_01$
```

Рисунок 2. Список запущенных процессов (конец).

PID (ID процесса) демона совпадает с PGID (ID группы процессов) и SID (ID сессии), т.к. этот процесс единственный в своей группе и сессии, а также является их лидером. TTY (имя терминала) – ?, т.к. процесс не имеет управляющего терминала. Stat – SS, так как процесс находится в S – прерываемом сне и s – лидер сессии.

На рисунке 3 представлено содержимое syslog.

```
lab1: Демон запущен!
lab1: Проверка на многократный запуск!
lab1: Lock-файл открыт!
lab1: Записываем PID!
lab1: Записали PID!
lab1: Проверка пройдена!
lab1: Демон!
CRON[7318]: (root) CMD ([ -x /etc/init.d/anacron ] && if [ ! -d /run/systemd/system ]; then /usr/sbin/invoke-rc.d anacron start >/dev/null; fi
lab1: Демон!
lab1: message repeated 5 times: [ Демон!]
```

Рисунок 3. Содержимое syslog.

На рисунках 4-5 представлен список запущенных процессов после команды kill [id_демона].

```
dalai@dalai-Inspiron-5567:~/Desktop/BMSTU/labs/lab_01$ sudo kill 13081
[sudo] password for dalai:
dalai@dalai-Inspiron-5567:~/Desktop/BMSTU/labs/lab_01$ ps -ajx
   PPID     PID    PGID     SID TTY        TPGID STAT   UID   TIME COMMAND
      0       1       1       1 ?           -1 Ss       0   0:04 /sbin/init splash
      0       2       0       0 ?           -1 S        0   0:00 [kthreadd]
      2       3       0       0 ?           -1 I<       0   0:00 [rcu_gp]
      2       4       0       0 ?           -1 I<       0   0:00 [rcu_par_gp]
      2       6       0       0 ?           -1 I<       0   0:00 [kworker/0:0H-events_h
      2       9       0       0 ?           -1 I<       0   0:00 [mm_percpu_wq]
      2      10       0       0 ?           -1 S        0   0:00 [rcu_tasks_rude_]
      2      11       0       0 ?           -1 S        0   0:00 [rcu_tasks_trace]
      2      12       0       0 ?           -1 S        0   0:00 [ksoftirqd/0]
      2      13       0       0 ?           -1 I        0   0:18 [rcu_sched]
      2      14       0       0 ?           -1 S        0   0:00 [migration/0]
      2      15       0       0 ?           -1 S        0   0:00 [idle_inject/0]
      2      16       0       0 ?           -1 S        0   0:00 [cpuhp/0]
      2      17       0       0 ?           -1 S        0   0:00 [cpuhp/1]
      2      18       0       0 ?           -1 S        0   0:00 [idle_inject/1]
      2      19       0       0 ?           -1 S        0   0:00 [migration/1]
      2      20       0       0 ?           -1 S        0   0:00 [ksoftirqd/1]
      2      22       0       0 ?           -1 I<       0   0:00 [kworker/1:0H-events_h
      2      23       0       0 ?           -1 S        0   0:00 [cpuhp/2]
      2      24       0       0 ?           -1 S        0   0:00 [idle_inject/2]
      2      25       0       0 ?           -1 S        0   0:00 [migration/2]
      2      26       0       0 ?           -1 S        0   0:00 [ksoftirqd/2]
      2      28       0       0 ?           -1 I<       0   0:00 [kworker/2:0H-events_h
```

Рисунок 4. Список запущенных процессов (начало).

```
  11564   11654   11564   11564 ?             -1 Sl    1000   0:15 /snap/code/82/usr/shar
  11654   11672   11564   11564 ?             -1 Sl    1000   0:15 /snap/code/82/usr/shar
  11672   11725   11725   11725 pts/2      11725 Ss+   1000   0:00 /usr/bin/bash
  11642   11920   11564   11564 ?             -1 Sl    1000   0:20 /home/dalai/.vscode/ex
  11920   11949   11564   11564 ?             -1 Sl    1000   0:05 /home/dalai/.vscode/ex
   5664   13145    1847    1847 ?             -1 Sl    1000   0:18 /opt/google/chrome/chr
      2   15025       0       0 ?             -1 I        0   0:01 [kworker/3:1-events]
   1588   15047    1599    1599 ?             -1 Sl    1000   0:02 evince /home/dalai/Des
      2   16067       0       0 ?             -1 D<       0   0:02 [kworker/u9:2+i915_fli
      2   16383       0       0 ?             -1 I        0   0:01 [kworker/u8:28-events_
      2   16398       0       0 ?             -1 I        0   0:01 [kworker/u8:43-i915]
      2   16413       0       0 ?             -1 S        0   0:00 [irq/128-mei_me]
      2   18438       0       0 ?             -1 I        0   0:00 [kworker/3:2-inet_frag
      2   18538       0       0 ?             -1 I        0   0:00 [kworker/2:1-events]
   5664   18571    1847    1847 ?             -1 Sl    1000   0:00 /opt/google/chrome/chr
      2   18740       0       0 ?             -1 I        0   0:00 [kworker/u8:0-events_p
      2   18883       0       0 ?             -1 I        0   0:02 [kworker/1:1-events]
      2   18885       0       0 ?             -1 I        0   0:00 [kworker/0:2-events]
   1588   19011    1945    1945 ?             -1 Sl    1000   0:02 gnome-control-center
      2   19113       0       0 ?             -1 I        0   0:00 [kworker/0:1-mm_percpu
      2   19218       0       0 ?             -1 I        0   0:00 [kworker/1:2-events]
      2   19239       0       0 ?             -1 I        0   0:00 [kworker/2:0-events]
      2   19313       0       0 ?             -1 I<       0   0:00 [kworker/u9:1]
      2   19329       0       0 ?             -1 I        0   0:00 [kworker/2:2-events]
      2   19344       0       0 ?             -1 I        0   0:00 [kworker/1:0-events]
      2   19386       0       0 ?             -1 I        0   0:00 [kworker/3:0-events]
  11300   19519   19519   11300 pts/0      19519 R+    1000   0:00 ps -ajx
dalai@dalai-Inspiron-5567:~/Desktop/BMSTU/labs/lab_01$
```

Рисунок 5. Список запущенных процессов (конец).