

Міністерство освіти та науки України
Одеський національний політехнічний університет
Інститут комп'ютерних систем
Кафедра інформаційних систем

Лабораторна робота №12
За дисципліною: “Операційні системи”
Тема: «Програмування міжпроцесної та багатопоточної
взаємодії»

Виконала
Студентка групи - АІ 202
Лобода Олександра

Мета роботи: вивчити особливості обміну інформацією між процесами за допомогою іменованих каналів, керування потоками, а також синхронізацію процесів через семафори та м'ютекси.

Задание 1

```
[root login: tue may 20 00:55:02 2021 from 195.130.02.200
[loboda_oleksandra@vpsj3IeQ ~]$ mkfifo loboda
[loboda_oleksandra@vpsj3IeQ ~]$ chmod u=rw loboda
[loboda_oleksandra@vpsj3IeQ ~]$ ls /etc > loboda
[loboda_oleksandra@vpsj3IeQ ~]$ █
```

```
[loboda_oleksandra@vpsj3IeQ ~]$ cat loboda
adjtime
aliases
aliases.db
alternatives
anacrontab
asound.conf
audisp
audit
bash_completion.d
bashrc
binfmt.d
centos-release
centos-release-upstream
chkconfig.d
chrony.conf
chrony.keys
cron.d
cron.daily
cron.deny
cron.hourly
cron.monthly
crontab
cron.weekly
crypttab
csh.cshrc
csh.login
dbus-1
dconf
default
depmod.d
```

```
[loboda_oleksandra@vpsj3IeQ ~]$ ls /etc/* >loboda
[loboda_oleksandra@vpsj3IeQ ~]$ █
```

```
[loboda_oleksandra@vpsj3IeQ ~]$ cat loboda
/etc/ld.so.cache
/etc/ld.so.conf
/etc/libaudit.conf
/etc/libuser.conf
/etc/localesdomains
/etc/locale.conf
/etc/localtime
/etc/login.defs
/etc/logrotate.conf
/etc/lynx.cfg
/etc/lynx.lss
/etc/lynx-site.cfg
[loboda_oleksandra@vpsj3IeQ ~]$ gzip -c < loboda > fileloboda.gz
[loboda_oleksandra@vpsj3IeQ ~]$ cat /etc/passwd > loboda
```

Задание 2

```

#include <sys/stat.h>
#include <fcntl.h>
#include <string.h>
#include <stdio.h>

#define NAMEDPIPE_NAME "loboda1"
#define BUFSIZE      50

int main (int argc, char ** argv) {
    int fd, len;
    char buf[BUFSIZE];

    if ( mkfifo(NAMEDPIPE_NAME, 0777) ) {
        fprintf(stderr,"Error in mkfifo!");
        return 1;
    }
    printf("%s is created\n", NAMEDPIPE_NAME);

    if ( (fd = open(NAMEDPIPE_NAME, O_RDONLY)) <= 0 ) {
        fprintf(stderr,"Error in open!");
        return 1;
    }
    printf("%s is opened\n", NAMEDPIPE_NAME);

    do {
        memset(buf, '\0', BUFSIZE);
        if ( (len = read(fd, buf, BUFSIZE-1)) <= 0 ) {
            printf("END!");
            close(fd);
            remove(NAMEDPIPE_NAME);
            return 0;
        }
        printf("Incomming message (%d): %s\n", len, buf);
    } while ( 1 );
}

```

```

lobodal is created
lobodal is opened
Incomming message (49): adjtime
aliases
aliases.db
alternatives
anacrontha
Incomming message (49): b
asound.conf
audisp
audit
bash_completion.d
bash
Incomming message (49): rc

```

```
[loboda_oleksandra@vpsj3IeQ ~]$ ls /etc > lobodal
END! [loboda_oleksandra@vpsj3IeQ ~]$ ./fifo
lobodal is created
lobodal is opened
Incomming message (49): /etc/ld.so.cache
/etc/ld.so.conf
/etc/libaudit.conf
Incomming message (49): nf
/etc/libuser.conf
/etc/localdomains
/etc/local
Incomming message (49): e.conf
/etc/localtime
/etc/login.defs
/etc/logrot
Incomming message (49): ate.conf
/etc/lynx.cfg
/etc/lynx.lss
/etc/lynx-si
```

```
[loboda_oleksandra@vpsj3IeQ ~]$ ls /etc/l* > lobodal
```

```
|END! [loboda_oleksandra@vpsj3IeQ ~]$ gzip -c < ./fifo > filelobodal.gz
```

```
[loboda_oleksandra@vpsj3IeQ ~]$ cat /etc/passwd > lobodal
```

```
filelobodal.gz | 2858 | May 25 14:14 ||
```

Задание 3

```

// компіляція з підключенням бібліотеки -lpthread

#include <stdio.h>
#include <pthread.h>

main() {
    pthread_t f2_thread, f1_thread;
    void *f2(), *f1();
    int i1 = 10, i2 = 10;
    pthread_create(&f1_thread, NULL, f1, &i1);
    pthread_create(&f2_thread, NULL, f2, &i2);
    pthread_join(f1_thread, NULL);
    pthread_join(f2_thread, NULL);
}

void *f1(int *x) {
    int i,n;
    n = *x;
    for (i=1;i<n;i++) {
        printf("Loboda");
        sleep(1);
    }
    pthread_exit(0);
}

void *f2(int *x) {
    int i,n;
    n = *x;
    for (i=1;i<n;i++) {
        printf("Oleksandra\n");
        sleep(1);
    }
    pthread_exit(0);
}

```

```

[loboda_oleksandra@vpsj3IeQ ~]$ gcc thread_1.c -o thread_1 -lpthread
[loboda_oleksandra@vpsj3IeQ ~]$ ./thread_1
LobodaOleksandra

```

Задание 4

```

#include <fcntl.h>
#include <sys/stat.h>
#include <semaphore.h>
#include <stdio.h>

#define SEMAPHORE_NAME "/loboda_semaphore"

int main(int argc, char ** argv) {
    sem_t *sem;

    if ( argc != 2 ) {
        if ((sem = sem_open(SEMAPHORE_NAME, O_CREAT, 0777, 0)) == SEM_FAILED ) {
            fprintf(stderr,"sem_open error");
            return 1;
        }
        printf("sem_open. Semaphore is taken.\nWaiting for it to be dropped. Loboda\n");
        if (sem_wait(sem) < 0 )
            fprintf(stderr,"sem_wait error");
        if ( sem_close(sem) < 0 )
            fprintf(stderr,"sem_close error");
        return 0;
    }
    else {
        printf("Dropping semaphore... Loboda\n");
        if ( (sem = sem_open(SEMAPHORE_NAME, 0)) == SEM_FAILED ) {
            fprintf(stderr,"sem_open error");
            return 1;
        }
        sem_post(sem);
        printf("sem_post. Semaphore dropped. Loboda\n");
        return 0;
    }
}

```

```

[loboda_oleksandra@vpsj3IeQ ~]$ gcc semaphore.c -o semaphore -lpthread
[loboda_oleksandra@vpsj3IeQ ~]$ ./semaphore
sem_open. Semaphore is taken.
Waiting for it to be dropped. Loboda
Dropping semaphore... Loboda
sem_post. Semaphore dropped. Loboda

```

Висновок: вивчили особливості обміну інформацією між процесами за допомогою іменованих каналів, керування потоками, а також синхронізацію процесів через семафори та м`ютекси.