

Міністерство освіти і науки України
Державний університет «Одеська політехніка»

Інститут комп'ютерних систем

Кафедра інформаційних систем

Лабораторна робота №12
з дисципліни «Операційні Системи»

Тема: «Програмування міжпроцесної та багатопоточної взаємодії»

Виконав:

ст. гр. АІ-204

Колесник К. В.

Перевірив:

Блажко О. А.

Дрозд М. О.

Одеса – 2021

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

Завдання для виконання:

Завдання 1 Робота з іменованими каналами

1.1 В домашньому каталозі вашого користувача створіть іменований канал з використанням команди `mkfifo`:

- назва каналу співпадає з вашим прізвищем у транслітерації
- права доступу до каналу (можна лише читати та писати власнику).

1.2 Підключіть до іменованого каналу процес, який буде в нього писати за такими командами:

- отримати зміст каталогу `/etc`
- отримати назви файлів, які починаються з букви вашого прізвища у транслітерації.

1.3 Перейдіть до нового терміналу роботи з ОС Linux та створіть процес, який буде читати зі створеного раніше каналу.

1.4 Поверніться до 1-го терміналу та підключіть до іменованого каналу процес, який буде в нього писати, архівуючи файл командою `gzip -c < pipe > file1.gz` де `pipe` – назва вашого каналу, `file1.gz` – назва файлу, який буде створено в результаті архівації

1.5 Перейдіть до 2-го терміналу роботи з ОС Linux та створіть процес, який буде читати зі створеного раніше каналу, архівуючи файл `/etc/passwd`

Завдання 2 Програмування іменованих каналів

Повторіть попереднє завдання, але пункт 2.1.1 виконайте через програмування іменованого каналу за прикладом з рисунку 1.

Завдання 3 Програмування потоків

За прикладом з рисунку 2 розробіть програму керування потоками, в якій в повідомленнях буде вказано ваше прізвище латиницею.

Виконайте програму за вказаним прикладом.

Завдання 4 Програмування семафорів

За прикладом з рисунку 3 розробіть програму керування семафором, в якій в повідомленнях буде вказано ваше прізвище латиницею.

Виконайте програму в двох терміналах за вказаним прикладом

Хід роботи:

Завдання 1 Робота з іменованими каналами

1.1 В домашньому каталозі вашого користувача створено іменований канал з використанням команди `mkfifo`:

- назва каналу співпадає з вашим прізвищем у транслітерації (“kolesnik”)
- права доступу до каналу (можна лише читати та писати власнику).

```
kolesnik_kirilo@vpsj3IeQ:~
login as: kolesnik_kirilo
kolesnik_kirilo@91.219.60.189's password:
Last login: Tue May 18 20:41:44 2021 from 188.163.101.235
[kolesnik_kirilo@vpsj3IeQ ~]$ mkfifo kolesnik
[kolesnik_kirilo@vpsj3IeQ ~]$ chmod u=rw,g=,o= kolesnik
[kolesnik_kirilo@vpsj3IeQ ~]$ ls -l
total 1616
-rw-rw-r-- 1 kolesnik_kirilo kolesnik_kirilo      10 Mar 29 05:06 123
-rw-rw-r-- 1 kolesnik_kirilo kolesnik_kirilo    1080 Mar 23 22:51 1.csv
-rw-rw-r-- 1 kolesnik_kirilo kolesnik_kirilo      10 Mar 29 05:35 1testfileforlab
5
-rw-rw-r-- 1 kolesnik_kirilo kolesnik_kirilo    1171 Mar 24 03:31 2.csv
-rw-rw-r-- 1 kolesnik_kirilo kolesnik_kirilo    1416 Mar 24 03:54 3.csv
-rw-rw-r-- 1 kolesnik_kirilo kolesnik_kirilo    5365 Mar 24 04:24 accounts.csv
-rw-rw-r-- 1 kolesnik_kirilo kolesnik_kirilo         0 Apr  5 10:50 blank.txt
d-wx-wx--x 2 kolesnik_kirilo kolesnik_kirilo    4096 Apr  5 12:01 darkCatalog
-rw-rw-r-- 3 kolesnik_kirilo kolesnik_kirilo      25 Mar 14 21:56 hard_link_1
-rw-rw-r-- 3 kolesnik_kirilo kolesnik_kirilo      25 Mar 14 21:56 hard_link_2
prw----- 1 kolesnik_kirilo kolesnik_kirilo         0 May 24 01:34 kolesnik
```

1.2 Підключено до іменованого каналу процес, який буде в нього писати за такими командами:

- отримати зміст каталогу /etc
- отримати назви файлів, які починаються з букви прізвища у транслітерації ("k").

1.3 Перейдено до нового терміналу роботи з ОС Linux та створено процес, який буде читати зі створеного раніше каналу.

```
kolesnik_kirilo@vpsj3IeQ:~
[kolesnik_kirilo@vpsj3IeQ ~]$ ls /etc | grep "^k" > kolesnik
[kolesnik_kirilo@vpsj3IeQ ~]$

kolesnik_kirilo@vpsj3IeQ:~
login as: kolesnik_kirilo
kolesnik_kirilo@91.219.60.189's password:
Last login: Mon May 24 01:32:06 2021 from 188.130.176.60
[kolesnik_kirilo@vpsj3IeQ ~]$ cat kolesnik
kdump.conf
kernel
krb5.conf
krb5.conf.d
kshrc
[kolesnik_kirilo@vpsj3IeQ ~]$
```

1.4 У 1-му терміналі підключено до іменованого каналу процес, який буде в нього писати, архівуючи файл командою `gzip -c < kolesnik > file1.gz` де `kolesnik` – назва каналу, `file1.gz` – назва файлу, який буде створено в результаті архівації

```
kolesnik_kirilo@vpsj3IeQ:~  
[kolesnik_kirilo@vpsj3IeQ ~]$ gzip -c < kolesnik > file1.gz
```

1.5 У 2-му терміналі роботи з ОС Linux створено процес, який буде читати зі створеного раніше каналу, архівуючи файл /etc/passwd

```
kolesnik_kirilo@vpsj3IeQ:~  
[kolesnik_kirilo@vpsj3IeQ ~]$ gzip -c < kolesnik > file1.gz  
[kolesnik_kirilo@vpsj3IeQ ~]$  
kolesnik_kirilo@vpsj3IeQ:~  
[kolesnik_kirilo@vpsj3IeQ ~]$ cat /etc/passwd > kolesnik | gunzip -c file1.gz  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin  
sync:x:5:0:sync:/sbin:/bin/sync  
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown  
halt:x:7:0:halt:/sbin:/sbin/halt  
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin  
operator:x:11:0:operator:/root:/sbin/nologin  
games:x:12:100:games:/usr/games:/sbin/nologin  
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin  
nobody:x:99:99:Nobody:/:/sbin/nologin  
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin  
dbus:x:81:81:System message bus:/:/sbin/nologin  
polkitd:x:999:997:User for polkitd:/:/sbin/nologin  
postfix:x:89:89:/var/spool/postfix:/sbin/nologin  
chrony:x:998:996:/var/lib/chrony:/sbin/nologin  
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin  
soft:x:1000:1000:/home/soft:/sbin/nologin  
saslauthd:x:997:76:Saslauthd user:/run/saslauthd:/sbin/nologin  
mailnull:x:47:47:/var/spool/mqueue:/sbin/nologin  
smmsp:x:51:51:/var/spool/mqueue:/sbin/nologin
```

Завдання 2 Програмування іменованих каналів

Повторно виконано попереднє завдання, але пункт 2.1.1 виконано через програмування іменованого каналу за прикладом з рисунку 1.

kolesnik_kirilo@vpsj3leQ:~

GNU nano 2.3.1

File: channel.c

Modified

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

#define NAMEDPIPE_NAME "kolesnik_2"
#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 );
}
```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

kolesnik_kirilo@vpsj3leQ:~

```
[kolesnik_kirilo@vpsj3leQ ~]$ gcc channel.c -o channel
[kolesnik_kirilo@vpsj3leQ ~]$ ./channel
kolesnik_2 is created
kolesnik_2 is opened
Incomming message (46): kdump.conf
kernel
krb5.conf
krb5.conf.d
kshrc
```

[kolesnik_kirilo@vpsj3leQ ~]\$

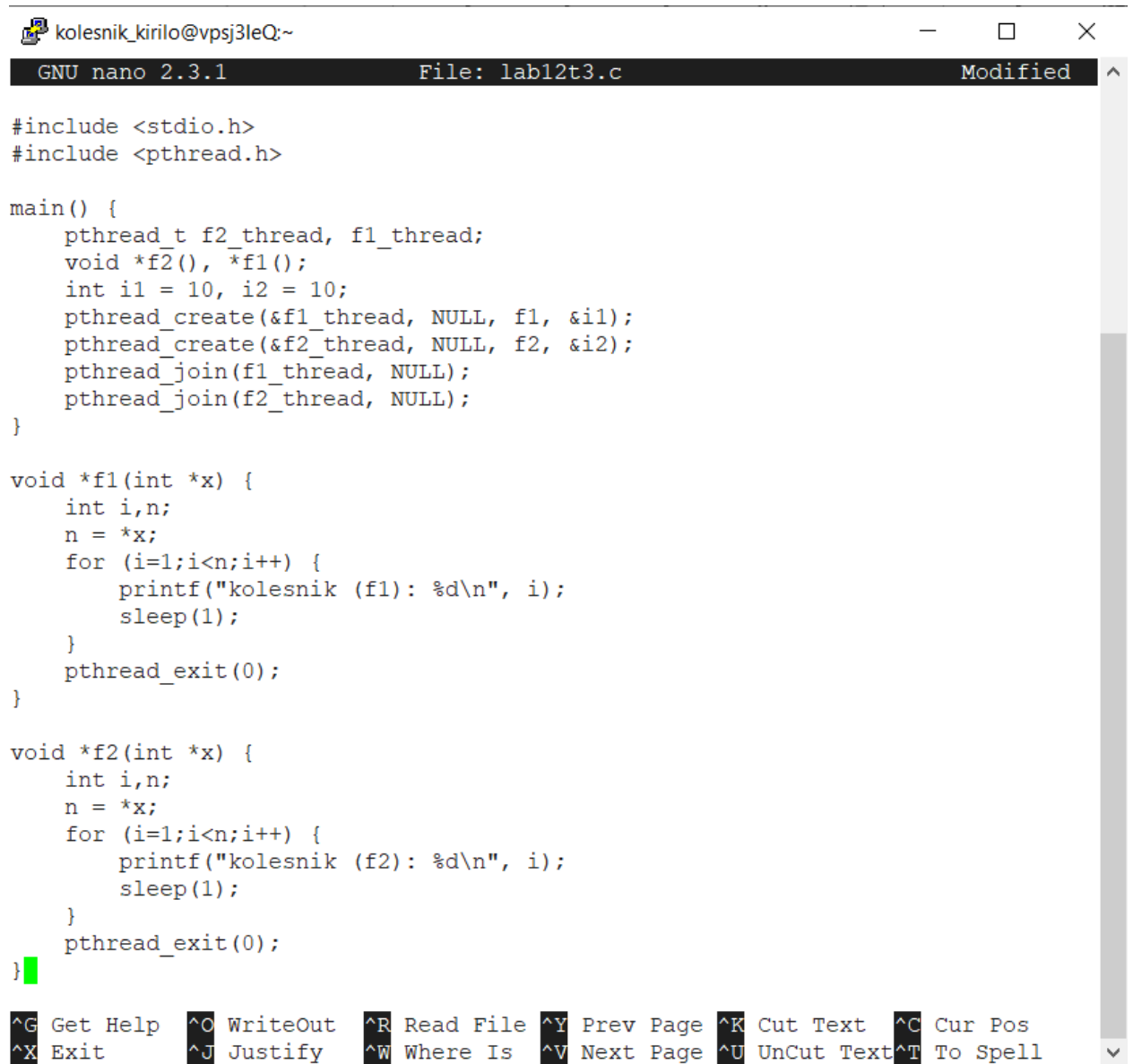
kolesnik_kirilo@vpsj3leQ:~

```
^ [kolesnik_kirilo@vpsj3leQ ~]$ ls /etc | grep "^k" > kolesnik_2
[kolesnik_kirilo@vpsj3leQ ~]$
```

Завдання 3 Програмування потоків

За прикладом з рисунку 2 розроблено програму керування потоками, в якій в повідомленнях буде вказано прізвище латиницею.

Виконано програму за вказаним прикладом.



```
kolesnik_kirilo@vpsj3leQ:~
GNU nano 2.3.1 File: lab12t3.c Modified ^

#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("kolesnik (f1): %d\n", i);
        sleep(1);
    }
    pthread_exit(0);
}

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

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

```
kolesnik_kirilo@vpsj3IeQ:~  
[kolesnik_kirilo@vpsj3IeQ ~]$ gcc lab12t3.c -o 112t3 -lpthread  
[kolesnik_kirilo@vpsj3IeQ ~]$ ./112t3  
kolesnik (f2): 1  
kolesnik (f1): 1  
kolesnik (f2): 2  
kolesnik (f1): 2  
kolesnik (f2): 3  
kolesnik (f1): 3  
kolesnik (f2): 4  
kolesnik (f1): 4  
kolesnik (f2): 5  
kolesnik (f1): 5  
kolesnik (f2): 6  
kolesnik (f1): 6  
kolesnik (f2): 7  
kolesnik (f1): 7  
kolesnik (f2): 8  
kolesnik (f1): 8  
kolesnik (f2): 9  
kolesnik (f1): 9  
[kolesnik_kirilo@vpsj3IeQ ~]$
```

Завдання 4 Програмування семафорів

За прикладом з рисунку 3 розроблено програму керування семафором, в якій в повідомленнях вказано прізвище латиницею.

Виконано програму в двох терміналах за вказаним прикладом

kolesnik_kirilo@vpsj3leQ:~

GNU nano 2.3.1

File: kolesemaaphore.c

Modified

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

#define SEMAPHORE_NAME "/kolesnik_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, "kolesnik sem_open error");
            return 1;
        }
        printf("sem_open. Kolesnik semaphore is taken.\nWaiting for it to be dro$
        if (sem_wait(sem) < 0 )
            fprintf(stderr, "kolesnik sem_wait error");
        if ( sem_close(sem) < 0 )
            fprintf(stderr, "kolesnik sem_close error");
        return 0;
    }
    else {
        printf("Dropping kolesnik semaphore...\n");
        if ( (sem = sem_open(SEMAPHORE_NAME, 0)) == SEM_FAILED ) {
            fprintf(stderr, "kolesnik sem_open error");
            return 1;
        }
        sem_post(sem);
        printf("sem_post. Kolesnik semaphore dropped.\n");
        return 0;
    }
}
```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

kolesnik_kirilo@vpsj3leQ:~

```
[kolesnik_kirilo@vpsj3leQ ~]$ nano kolesemaaphore.c
[kolesnik_kirilo@vpsj3leQ ~]$ gcc kolesemaaphore.c -o kolesema -lpthread
[kolesnik_kirilo@vpsj3leQ ~]$ ./kolesema
sem_open. Kolesnik semaphore is taken.
Waiting for it to be dropped.
[kolesnik_kirilo@vpsj3leQ ~]$
```

kolesnik_kirilo@vpsj3leQ:~

```
[kolesnik_kirilo@vpsj3leQ ~]$ ./kolesema 1
Dropping Kolesnik semaphore...
sem_post. Kolesnik semaphore dropped.
[kolesnik_kirilo@vpsj3leQ ~]$
```

Висновок: Під час виконання цієї лабораторної роботи, було на практиці розглянуто принципи взаємодії типу “клієнт – сервер” із детальним дослідженням особливостей обміну інформацією між процесами за допомогою іменованих каналів, керування потоками, а також синхронізації процесів через семафори та м'ютекси. Виявлено, що користувач отримує можливість керувати потоками та здійснювати обмін інформацією між процесами та синхронізувати їх, користуючись семафорами та м'ютексами та засобами мови програмування C. Складностей під час виконання роботи не виникало.