

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

Лабораторна робота №12  
З дисципліни "Операційні системи"  
Тема : «Програмування міжпроцесної та багатопотокової взаємодії»

Виконав  
Ст.гр. АІ-204  
Костецький Б.В.  
Перевірили:  
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**

### **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*

### **2.Програмування іменованих каналів**

Повторіть попереднє завдання, але пункт 2.1 виконайте через програмування іменованого каналу за прикладом з рисунку 1.

### **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.

### **4.Програмування семафорів**

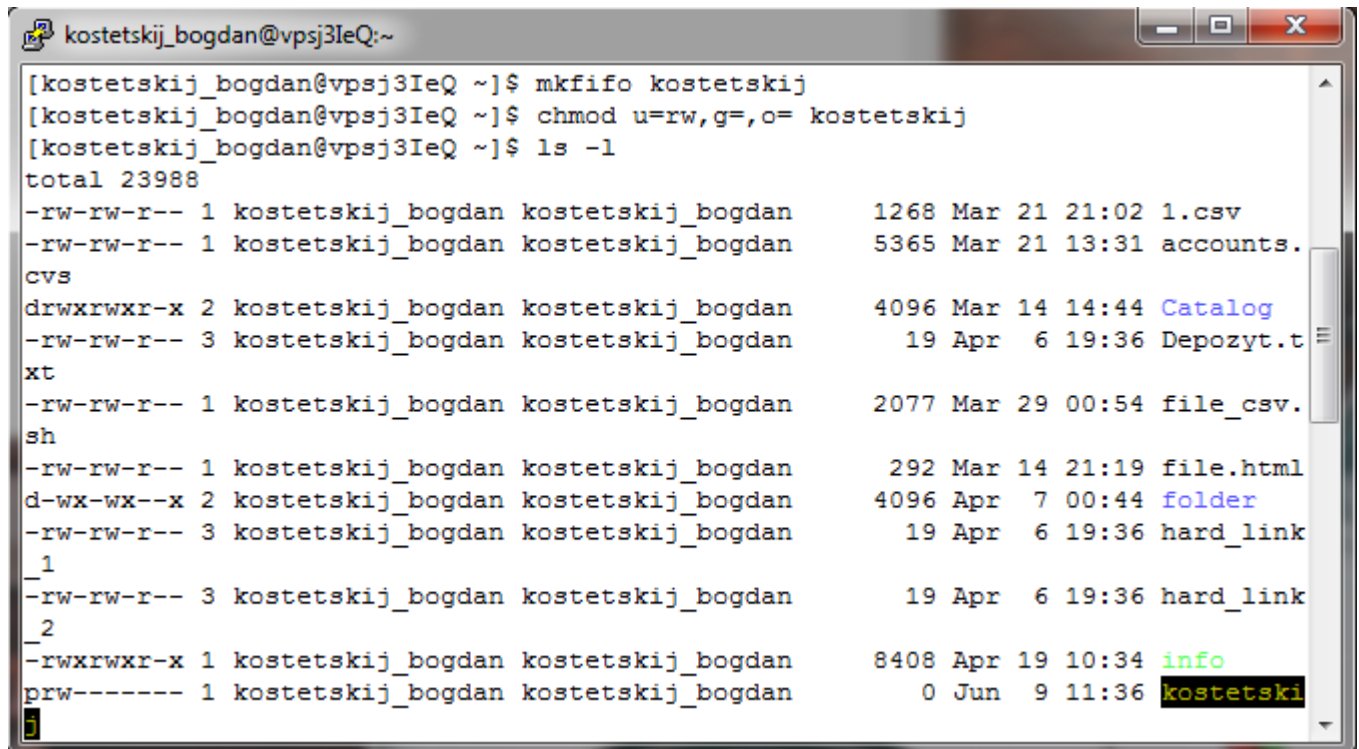
За прикладом з рисунку 3 розробіть програму керування семафором, в якій в повідомленнях буде вказано ваше прізвище латиницею.

Виконайте програму в двох терміналах за вказаним прикладом.

## Хід роботи:

### Завдання 1

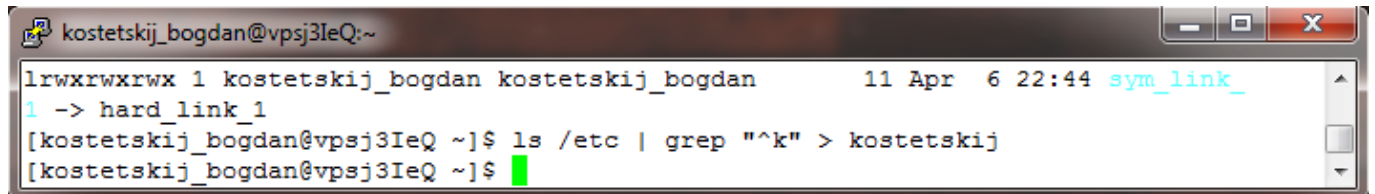
1. В домашньому каталозі користувача створили іменованний канал з використанням команди `mkfifo`



```
kostetskij_bogdan@vpsj3IeQ:~  
[kostetskij_bogdan@vpsj3IeQ ~]$ mkfifo kostetskij  
[kostetskij_bogdan@vpsj3IeQ ~]$ chmod u=rw,g=,o= kostetskij  
[kostetskij_bogdan@vpsj3IeQ ~]$ ls -l  
total 23988  
-rw-rw-r-- 1 kostetskij_bogdan kostetskij_bogdan 1268 Mar 21 21:02 1.csv  
-rw-rw-r-- 1 kostetskij_bogdan kostetskij_bogdan 5365 Mar 21 13:31 accounts.  
cvs  
drwxrwxr-x 2 kostetskij_bogdan kostetskij_bogdan 4096 Mar 14 14:44 Catalog  
-rw-rw-r-- 3 kostetskij_bogdan kostetskij_bogdan 19 Apr 6 19:36 Depozyt.t  
xt  
-rw-rw-r-- 1 kostetskij_bogdan kostetskij_bogdan 2077 Mar 29 00:54 file_csv.  
sh  
-rw-rw-r-- 1 kostetskij_bogdan kostetskij_bogdan 292 Mar 14 21:19 file.html  
d-wx-wx--x 2 kostetskij_bogdan kostetskij_bogdan 4096 Apr 7 00:44 folder  
-rw-rw-r-- 3 kostetskij_bogdan kostetskij_bogdan 19 Apr 6 19:36 hard_link  
_1  
-rw-rw-r-- 3 kostetskij_bogdan kostetskij_bogdan 19 Apr 6 19:36 hard_link  
_2  
-rwxrwxr-x 1 kostetskij_bogdan kostetskij_bogdan 8408 Apr 19 10:34 info  
prw----- 1 kostetskij_bogdan kostetskij_bogdan 0 Jun 9 11:36 kostetski  
j
```

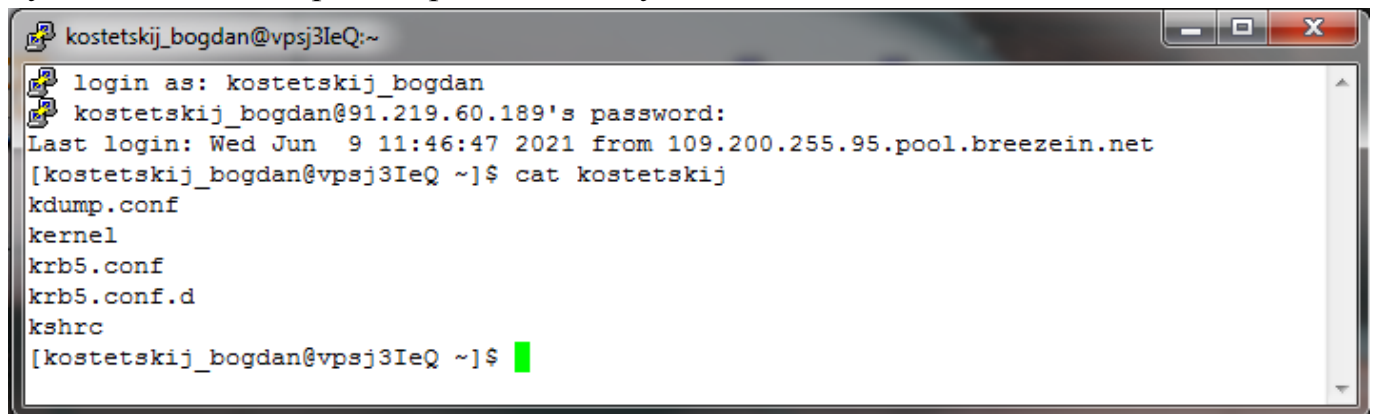
2. Підключили до іменованого каналу процес, який буде в нього писати за такими командами:

- отримати зміст каталогу `/etc`
- отримати назви файлів, які починаються з букви 'k'.



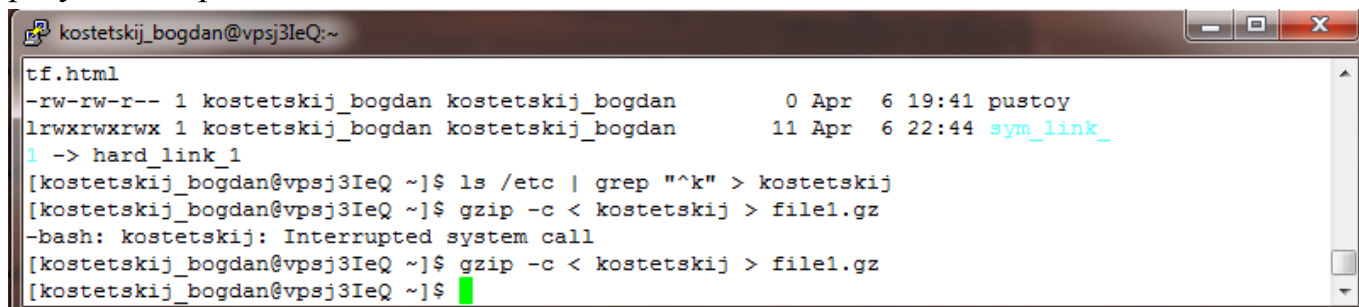
```
kostetskij_bogdan@vpsj3IeQ:~  
lrwxrwxrwx 1 kostetskij_bogdan kostetskij_bogdan 11 Apr 6 22:44 sym_link_  
1 -> hard_link 1  
[kostetskij_bogdan@vpsj3IeQ ~]$ ls /etc | grep "^k" > kostetskij  
[kostetskij_bogdan@vpsj3IeQ ~]$
```

3. Перейшли до нового терміналу роботи з ОС Linux та створили процес, який буде читати зі створеного раніше каналу.



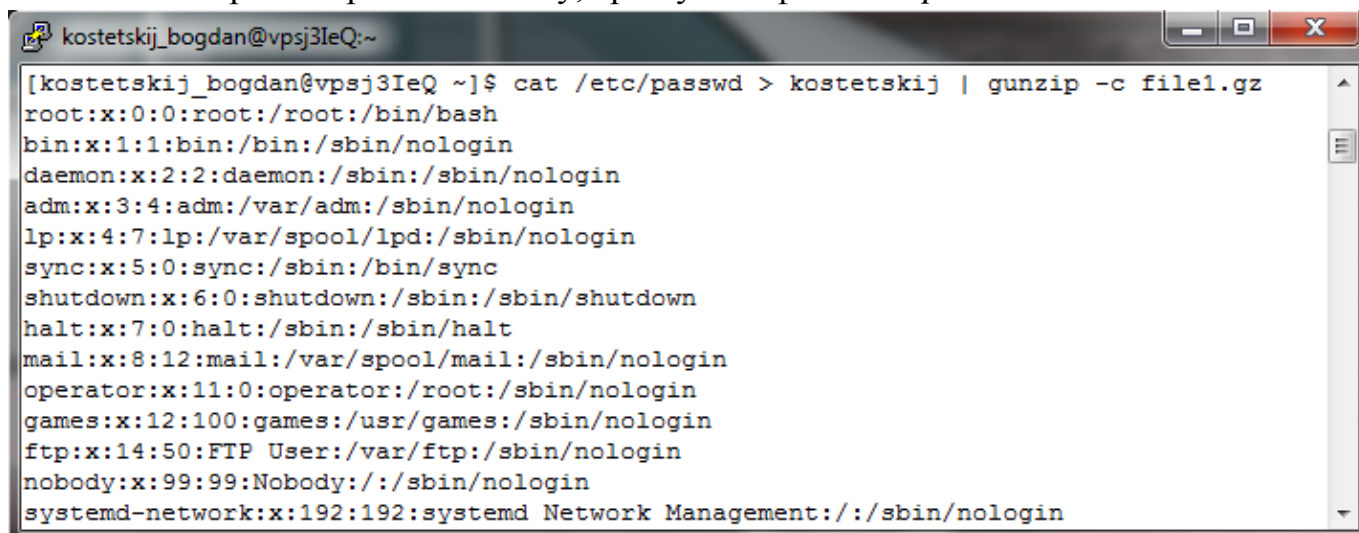
```
kostetskij_bogdan@vpsj3IeQ:~  
login as: kostetskij_bogdan  
kostetskij_bogdan@91.219.60.189's password:  
Last login: Wed Jun 9 11:46:47 2021 from 109.200.255.95.pool.breezein.net  
[kostetskij_bogdan@vpsj3IeQ ~]$ cat kostetskij  
kdump.conf  
kernel  
krb5.conf  
krb5.conf.d  
kshrc  
[kostetskij_bogdan@vpsj3IeQ ~]$
```

4. Повернулися до 1-го терміналу та підключили 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



```
kostetskij_bogdan@vpsj3IeQ:~  
tf.html  
-rw-rw-r-- 1 kostetskij_bogdan kostetskij_bogdan      0 Apr  6 19:41 pustoy  
lrwxrwxrwx 1 kostetskij_bogdan kostetskij_bogdan      11 Apr  6 22:44 sym_link_  
1 -> hard_link_1  
[kostetskij_bogdan@vpsj3IeQ ~]$ ls /etc | grep "^k" > kostetskij  
[kostetskij_bogdan@vpsj3IeQ ~]$ gzip -c < kostetskij > file1.gz  
-bash: kostetskij: Interrupted system call  
[kostetskij_bogdan@vpsj3IeQ ~]$ gzip -c < kostetskij > file1.gz  
[kostetskij_bogdan@vpsj3IeQ ~]$
```

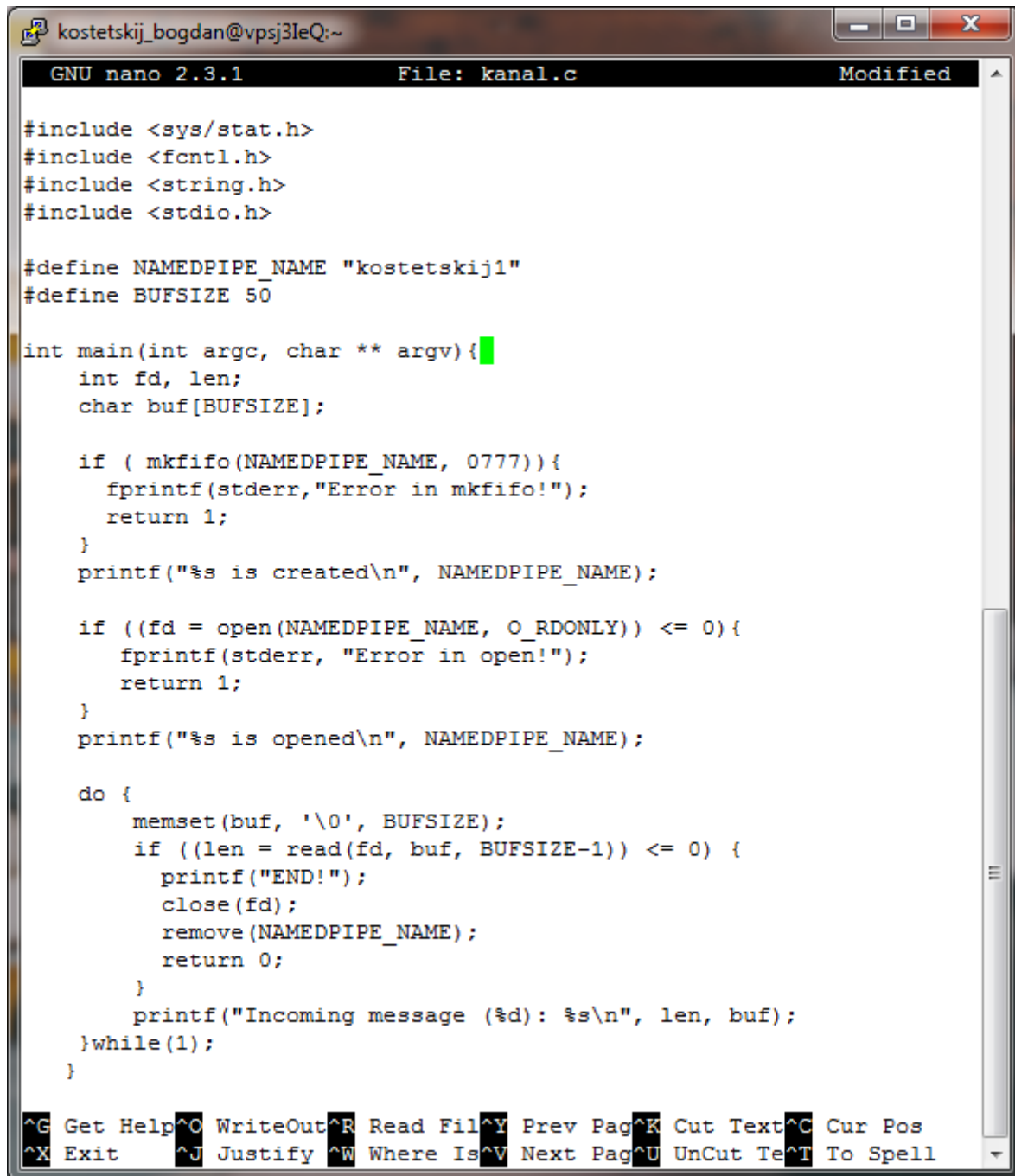
5. Перейшли до 2-го терміналу роботи з ОС Linux та створили процес, який буде читати зі створеного раніше каналу, архівуючи файл `/etc/passwd`



```
kostetskij_bogdan@vpsj3IeQ:~  
[kostetskij_bogdan@vpsj3IeQ ~]$ cat /etc/passwd > kostetskij | 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
```

## Завдання 2

Повторили попереднє завдання, але пункт 2.1 виконали через програмування іменованого каналу за прикладом з рисунку 1.



```
GNU nano 2.3.1      File: kanal.c      Modified

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

#define NAMEDPIPE_NAME "kostetskij1"
#define BUFSIZE 50

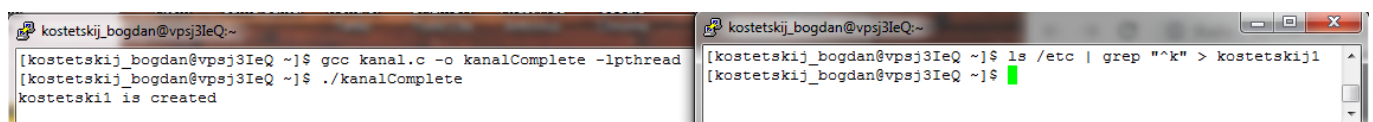
int main(int argc, char ** argv){
    int fd, len;
    char buf[BUFSIZE];

    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("Incoming 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

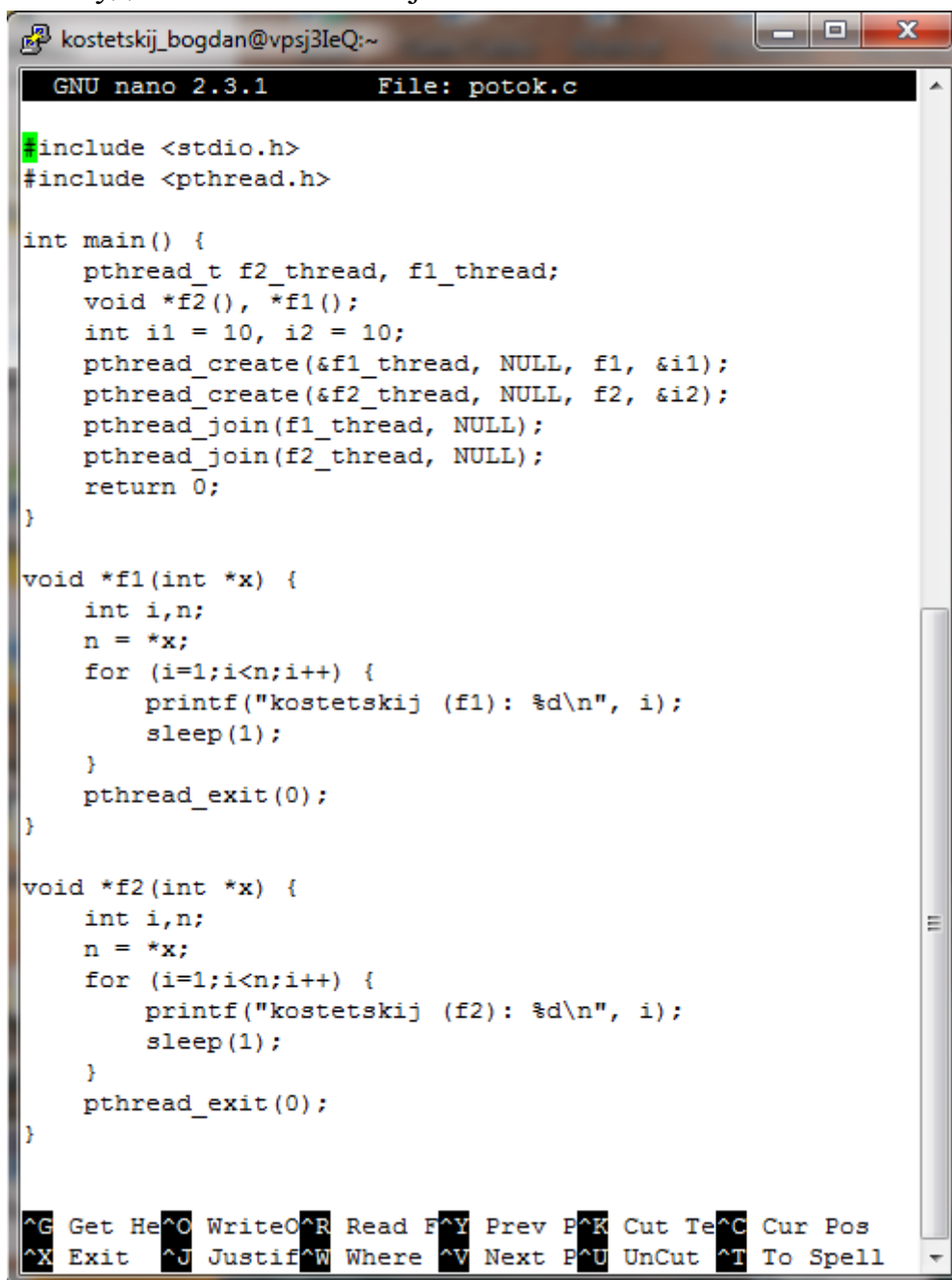


```
kostetskij_bogdan@vpsj3IeQ:~$ gcc kanal.c -o kanalComplete -lpthread
kostetskij_bogdan@vpsj3IeQ:~$ ./kanalComplete
kostetskij1 is created

kostetskij_bogdan@vpsj3IeQ:~$ ls /etc | grep "^k" > kostetskij1
kostetskij_bogdan@vpsj3IeQ:~$
```

### Завдання 3

За прикладом з рисунку 2 розробили програму керування потоками, в якій в повідомленнях буде вказано kostetskij.



```
kostetskij_bogdan@vpsj3leQ:~
GNU nano 2.3.1      File: potok.c

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

int 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);
    return 0;
}

void *f1(int *x) {
    int i,n;
    n = *x;
    for (i=1;i<n;i++) {
        printf("kostetskij (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("kostetskij (f2): %d\n", i);
        sleep(1);
    }
    pthread_exit(0);
}

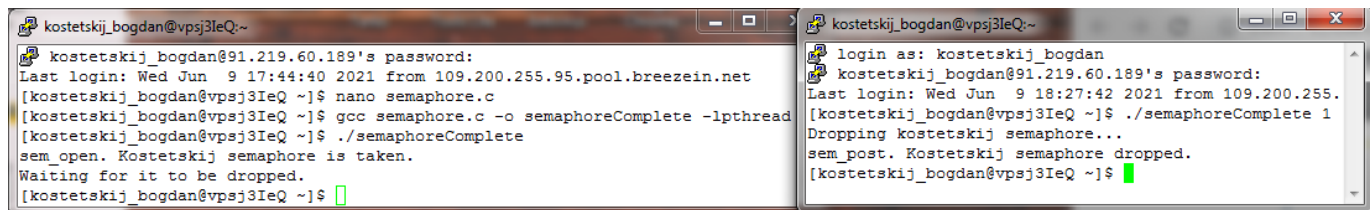
^G Get He^O WriteO^R Read F^Y Prev P^K Cut Te^C Cur Pos
^X Exit  ^J Justif^W Where ^V Next P^U UnCut ^T To Spell
```

```
kostetskij_bogdan@vpsj3IeQ:~  
[kostetskij_bogdan@vpsj3IeQ ~]$ nano potok.c  
[kostetskij_bogdan@vpsj3IeQ ~]$ gcc potok.c -o potokComplete -lpthread  
[kostetskij_bogdan@vpsj3IeQ ~]$ ./potokComplete  
kostetskij (f1): 1  
kostetskij (f2): 1  
kostetskij (f2): 2  
kostetskij (f1): 2  
kostetskij (f2): 3  
kostetskij (f1): 3  
kostetskij (f2): 4  
kostetskij (f1): 4  
kostetskij (f2): 5  
kostetskij (f1): 5  
kostetskij (f2): 6  
kostetskij (f1): 6  
kostetskij (f2): 7  
kostetskij (f1): 7  
kostetskij (f2): 8  
kostetskij (f1): 8  
kostetskij (f2): 9  
kostetskij (f1): 9  
[kostetskij_bogdan@vpsj3IeQ ~]$
```

#### Завдання 4

За прикладом з рисунку 3 розробили програму керування семафором, в якій в повідомленнях буде вказано kostetskij.

```
GNU nano 2.3.1      File: semaphore.c  
  
#include <fcntl.h>  
#include <sys/stat.h>  
#include <semaphore.h>  
#include <stdio.h>  
  
#define SEMAPHORE_NAME "/kostetskij_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, "kostetskij sem_open error");  
            return 1;  
        }  
        printf("sem_open. Kostetskij semaphore is taken.\nWaiting for it to be dropped.$  
if (sem_wait(sem) < 0 )  
        fprintf(stderr, "kostetskij sem_wait error");  
        if ( sem_close(sem) < 0 )  
            fprintf(stderr, "kostetskij sem_close error");  
        return 0;  
    }  
    else {  
        printf("Dropping kostetskij semaphore...\n");  
        if ( (sem = sem_open(SEMAPHORE_NAME, 0)) == SEM_FAILED ) {  
            fprintf(stderr, "kostetskij sem_open error");  
            return 1;  
        }  
        sem_post(sem);  
        printf("sem_post. Kostetskij 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
```



The image shows two terminal windows side-by-side. The left window shows a user logging in, creating a semaphore file named 'semaphore.c', compiling it with 'gcc semaphore.c -o semaphoreComplete -lpthread', and then running './semaphoreComplete'. The output shows 'sem\_open. Kostetskij semaphore is taken.' and 'Waiting for it to be dropped.' The right window shows the same user logging in again, then running './semaphoreComplete 1'. The output shows 'Dropping kostetskij semaphore...', 'sem\_post. Kostetskij semaphore dropped.', and the prompt returns.

```
kostetskij_bogdan@vpsj3IeQ:~  
kostetskij_bogdan@91.219.60.189's password:  
Last login: Wed Jun  9 17:44:40 2021 from 109.200.255.95.pool.breezein.net  
[kostetskij_bogdan@vpsj3IeQ ~]$ nano semaphore.c  
[kostetskij_bogdan@vpsj3IeQ ~]$ gcc semaphore.c -o semaphoreComplete -lpthread  
[kostetskij_bogdan@vpsj3IeQ ~]$ ./semaphoreComplete  
sem_open. Kostetskij semaphore is taken.  
Waiting for it to be dropped.  
[kostetskij_bogdan@vpsj3IeQ ~]$  
  
kostetskij_bogdan@vpsj3IeQ:~  
login as: kostetskij_bogdan  
kostetskij_bogdan@91.219.60.189's password:  
Last login: Wed Jun  9 18:27:42 2021 from 109.200.255.  
[kostetskij_bogdan@vpsj3IeQ ~]$ ./semaphoreComplete 1  
Dropping kostetskij semaphore...  
sem_post. Kostetskij semaphore dropped.  
[kostetskij_bogdan@vpsj3IeQ ~]$
```

**Висновок:** в ході виконання лабораторної роботи , було вивчено особливості обміну інформацією між процесами за допомогою іменованих каналів, керування потоками, а також синхронізацію процесів через семафори та м'ютекси.