

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:  
Студент групи AI-205  
Кучеренко М.М.

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 до виконання**

### **2.1 Робота з іменованими каналами**

2.1.1 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).

2.1.2 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.

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

2.1.4 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

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

### **2.2 Програмування іменованих каналів**

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

### **2.3 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.

### **2.4 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.

## Скріншоти виконання завдань:

### Завдання 2.1.1:

```
kucherenko_maksim@vpsj3IeQ:~  
[kucherenko_maksim@vpsj3IeQ ~]$ mkfifo kucherenko  
[kucherenko_maksim@vpsj3IeQ ~]$ chmod u=rx,g=,o= kucherenko  
[kucherenko_maksim@vpsj3IeQ ~]$ ls -l  
total 7904  
-rw-rw-r-- 1 kucherenko_maksim kucherenko_maksim 5365 Mar 26 18:57 accounts.csv  
-rw-rw-r-- 1 kucherenko_maksim kucherenko_maksim 0 Apr 5 00:58 filename_filename1234  
pr-x----- 1 kucherenko_maksim kucherenko_maksim 0 May 30 23:55 kucherenko  
lrwxrwxrwx 1 kucherenko_maksim kucherenko_maksim 13 Apr 14 03:16 kucherenko  
2.sh -> kucherenko.sh  
lrwxrwxrwx 1 kucherenko_maksim kucherenko_maksim 13 Apr 14 03:16 kucherenko  
3.sh -> kucherenko.sh  
-rw-rw-r-- 1 kucherenko_maksim kucherenko_maksim 2984 Mar 27 18:12 kucherenko  
7.csv  
-rw-rw-r-- 1 kucherenko_maksim kucherenko_maksim 2908 Mar 27 18:55 kucherenko  
7_edited_1.csv  
-rw-rw-r-- 1 kucherenko_maksim kucherenko_maksim 2887 Mar 27 18:55 kucherenko  
7_edited_2.csv  
-rw-rw-r-- 1 kucherenko_maksim kucherenko_maksim 1195 Mar 27 19:04 kucherenko  
7_edited_3.csv  
-rw-rw-r-- 1 kucherenko_maksim kucherenko_maksim 2984 Mar 27 18:55 kucherenko  
7_edited.csv  
drwxrwxr-x 2 kucherenko_maksim kucherenko_maksim 4096 Mar 17 01:30 kucherenko
```

### Завдання 2.1.2:

```
kucherenko_maksim@vpsj3IeQ:~  
[kucherenko_maksim@vpsj3IeQ ~]$ ls /etc | grep "^b" > kucherenko
```

### Завдання 2.1.3:

```
kucherenko_maksim@vpsj3IeQ:~  
[kucherenko_maksim@vpsj3IeQ ~]$ ls /etc | grep "^b" > kucherenko  
[kucherenko_maksim@vpsj3IeQ ~]$  
kucherenko_maksim@vpsj3IeQ:~  
[kucherenko_maksim@vpsj3IeQ ~]$ cat kucherenko  
bash_completion.d  
bashrc  
binfmt.d  
[kucherenko_maksim@vpsj3IeQ ~]$
```

### Завдання 2.1.4:

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

### Завдання 2.1.5:

```
kucherenko_maksim@vpsj3leQ:~  
[kucherenko_maksim@vpsj3leQ ~]$ gzip -c < kucherenko > file1.gz  
[kucherenko_maksim@vpsj3leQ ~]$
```

```
kucherenko_maksim@vpsj3leQ:~  
[kucherenko_maksim@vpsj3leQ ~]$ cat /etc/passwd > kucherenko | 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  
saslauth: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 2.2:

```
kucherenko_maksim@vpsj3leQ:~  
GNU nano 2.3.1 File: chanel.c  
  
#include <sys/stat.h>  
#include <fcntl.h>  
#include <string.h>  
#include <stdio.h>  
  
#define NAMEDPIPE_NAME "kucherenko_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.\n");  
        return 1;  
    }  
  
    printf("%s is created.\n", NAMEDPIPE_NAME);  
  
    if ((fd = open(NAMEDPIPE_NAME, O_RDONLY)) <= 0) {  
        fprintf(stderr, "Error in open!\n");  
        return 1;  
    }  
  
    printf("%s is opened\n", NAMEDPIPE_NAME);  
  
    do {  
        memset(buf, '\0', BUFSIZE);  
        if ((len = read(fd, buf, BUFSIZE - 1)) <= 0) {  
            printf("END!\n");  
            close(fd);  
            remove(NAMEDPIPE_NAME);  
            return 0;  
        }  
  
        printf("Incoming message (%d): %s\n", len, buf);  
    }  
  
    while(1);  
}
```

```
kucherenko_maksim@vpsj3IeQ:~  
[kucherenko_maksim@vpsj3IeQ ~]$ nano chanel.c  
[kucherenko_maksim@vpsj3IeQ ~]$ gcc -o chanel chanel.c  
[kucherenko_maksim@vpsj3IeQ ~]$ ./chanel  
kucherenko_2 is created.  
kucherenko_2 is opened  
Incoming message (34): bash_completion.d  
bashrc  
binfmt.d  
  
END!  
[kucherenko_maksim@vpsj3IeQ ~]$
```

```
kucherenko_maksim@vpsj3IeQ:~  
[kucherenko_maksim@vpsj3IeQ ~]$ ls /etc | grep "^b" > kucherenko_2  
[kucherenko_maksim@vpsj3IeQ ~]$
```

### Завдання 2.3:

```
kucherenko_maksim@vpsj3IeQ:~  
GNU nano 2.3.1 File: lab12_task2-3.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("Kucherenko (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("Kucherenko (f2): %d\n", i);  
        sleep(1);  
    }  
  
    pthread_exit(0);  
}
```


```
kucherenko_maksim@vpsj3IeQ:~  
[kucherenko_maksim@vpsj3IeQ ~]$ nano lab12_task2-3.c  
[kucherenko_maksim@vpsj3IeQ ~]$ gcc -o lab12_task2-3 lab12_task2-3.c -lpthread  
[kucherenko_maksim@vpsj3IeQ ~]$ ./lab12_task2-3  
Kucherenko (f2): 1  
Kucherenko (f1): 1  
Kucherenko (f2): 2  
Kucherenko (f1): 2  
Kucherenko (f2): 3  
Kucherenko (f1): 3  
Kucherenko (f2): 4  
Kucherenko (f1): 4  
Kucherenko (f2): 5  
Kucherenko (f1): 5  
Kucherenko (f2): 6  
Kucherenko (f1): 6  
Kucherenko (f2): 7  
Kucherenko (f1): 7  
Kucherenko (f2): 8  
Kucherenko (f1): 8  
Kucherenko (f2): 9  
Kucherenko (f1): 9  
[kucherenko_maksim@vpsj3IeQ ~]$ █
```

## Завдання 2.4:

```
kucherenko_maksim@vpsj3IeQ:~  
GNU nano 2.3.1 File: lab12_task2-4.c  
#include <fcntl.h>  
#include <sys/stat.h>  
#include <semaphore.h>  
#include <stdio.h>  
  
#define SEMAPHORE_NAME "/kucherenkos_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, "Kucherenkos sem_open error");  
            return 1;  
        }  
  
        printf("sem_open. Kucherenkos semaphore is taken.\nWaiting for it to be dropped.\n");  
        if (sem_wait(sem) < 0)  
            fprintf(stderr, "Kucherenkos sem_wait error.\n");  
        if (sem_close(sem) < 0)  
            fprintf(stderr, "Kucherenkos sem_close error.\n");  
        return 0;  
    }  
    else {  
        printf("Dropping Kucherenkos semaphore...\n");  
        if ((sem = sem_open(SEMAPHORE_NAME, 0)) == SEM_FAILED) {  
            fprintf(stderr, "Kucherenkos sem_open error.\n");  
            return 1;  
        }  
  
        sem_post(sem);  
        printf("sem_post. Kucherenkos semaphore dropped.\n");  
        return 0;  
    }  
}
```

 kucherenko\_maksim@vpsj3IeQ:~

```
[kucherenko_maksim@vpsj3IeQ ~]$ nano labal2_task2-4.c
[kucherenko_maksim@vpsj3IeQ ~]$ gcc -o labal2_task2-4 labal2_task2-4.c -lpthread
[kucherenko_maksim@vpsj3IeQ ~]$ ./labal2_task2-4
sem_open. Kucherenkos semaphore is taken.
Waiting for it to be dropped.
[kucherenko_maksim@vpsj3IeQ ~]$
```

 kucherenko\_maksim@vpsj3IeQ:~

```
[kucherenko_maksim@vpsj3IeQ ~]$ ./labal2_task2-4 1
Dropping Kucherenkos semaphore...
sem_post. Kucherenkos semaphore dropped.
[kucherenko_maksim@vpsj3IeQ ~]$
```

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