

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ  
УНІВЕРСИТЕТ

Інститут комп'ютерних систем

Кафедра інформаційних  
систем

Лабораторна робота №12

З дисципліни: «Операційні системи»

**Тема: «Програмування міжпроцесної та багатопоточної  
взаємодії»**

Виконала:  
Студентка групи  
AI203  
Веселкова А.С.

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 звикористанням 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 зтакими 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 2.1.1 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.

## Результати виконання завдань:

1. Створіть іменованний канал з використанням команди `mkfifo`  

```
prw----- 1 veselkova_anna veselkova_anna    0 May 23 18:30 veselkova
[veselkova_anna@vpsj3IeQ ~]$ mkfifo veselkova
[veselkova_anna@vpsj3IeQ ~]$ chmod g-rw, o-r veselkova
chmod: invalid mode: 'g-rw,'
Try 'chmod --help' for more information.
[veselkova_anna@vpsj3IeQ ~]$ chmod g-rw,o-r veselkova
[veselkova_anna@vpsj3IeQ ~]$ ls -l | grep 'veselkova'
```

Підключіть до іменованого каналу процес, який буде в нього писати. Створіть процес, який буде читати зі створеного раніше каналу

```
[veselkova_anna@vpsj3IeQ ~]$ ls /etc | grep '^g' > veselkova
[veselkova_anna@vpsj3IeQ ~]$ cat veselkova
gconf
gcrypt
ghostscript
glvnd
gnupg
groff
group
group-
grub2.cfg
grub.d
gshadow
gshadow-
gss
gssproxy
```

Підключіть до іменованого каналу процес, який буде в нього писати, архівуючи файл

```
[veselkova_anna@vpsj3IeQ ~]$ ls /etc | grep '^g' > veselkova | gzip -c <
veselkova > veselkova.gz
```

Створіть процес, який буде читати зі створеного раніше каналу, архівуючи файл

/etc/passwd

```
[veselkova_anna@vpsj3IeQ ~]$ cat veselkova | gzip -c </etc/passwd> passwd.gz
```

```

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

#define pipe_name "veselkova"
#define size      50

int main (int argc, char ** argv) {
    int fd, len;
    char buf[size];
    if ( mkfifo(pipe_name, 0777) ) {
        fprintf(stderr, "Error in mkfifo!\n");
        return 1;
    }
    printf("%s is created\n", pipe_name);
    if ( (fd = open(pipe_name, O_RDONLY)) <= 0 ) {
        fprintf(stderr, "Error in open!\n");
        return 1;
    }
    printf("%s is opened\n", pipe_name);
    do {
        memset(buf, '\0', size);
        if ( (len = read(fd, buf, size-1)) <= 0 ) {
            printf("END!");
            close(fd);
            remove(pipe_name);
            return 0;
        }
        printf("Incomming message (%d): %s\n", len, buf);
    } while ( 1 );
}

```

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

```

[veselkova_anna@vpsj3IeQ ~]$ gcc -o pipe pipe.c
[veselkova_anna@vpsj3IeQ ~]$ ./pipe
veselkova is created
veselkova is opened
Incomming message (49): gconf
gcrypt
ghostscript
glvnd
gnupg
groff
group

Incomming message (49): group-
grub2.cfg
grub.d
gshadow
gshadow-
gss
gssp
Incomming message (5): roxy

```

```
[veselkova_anna@vpsj3IeQ ~]$ ls /etc | grep '^g' > veselkova
```

```
[veselkova_anna@vpsj3IeQ ~]$ ~]$ ls /etc | grep '^g' > veselkova | gzip -c <  
veselkova > veselkova.gz
```

```
[veselkova_anna@vpsj3IeQ ~]$ ./pipe
```

veselkova is created

veselkova is opened

Incomming message (49): gconf

gcrypt

ghostscript

glvnd

gnupg

groff

group

Incomming message (49): group-

grub2.cfg

grub.d

gshadow

gshadow-

gss

gssp

Incomming message (5): roxy

```
[veselkova_anna@vpsj3IeQ ~]$ ls /etc | grep '^g' > veselkova | gzip -c <  
/etc/passwd> passwd.gz
```

```
[veselkova_anna@vpsj3IeQ ~]$ eQ ~]$ ./pipe
```

veselkova is created

veselkova is opened

Incomming message (49): gconf

gcrypt

ghostscript

glvnd

gnupg

groff

group

Incomming message (49): group-

grub2.cfg

grub.d

gshadow

gshadow-

gss

gssp

Incomming message (5): roxy

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

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

main() {
    pthread_t f2_thread, f1_thread;
    void *f2(), *f1();
    char s1[] = "veselkova", s2[] = "veselkova";
    pthread_create(&f1_thread, NULL, f1, &s1);
    pthread_create(&f2_thread, NULL, f2, &s2);
    pthread_join(f1_thread, NULL);
    pthread_join(f2_thread, NULL);
}

void *f1(char x[]) {
    printf("f1: %s\n", x);
    pthread_exit(0);
}

void *f2(int x[]) {
    printf("f2: %s\n", x);
    pthread_exit(0);
}

[veselkova_anna@vpsj3IeQ ~]$ gcc -o thread thread.c -lpthread
[veselkova_anna@vpsj3IeQ ~]$ ./thread
f2: veselkova
f1: veselkova
```

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

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

#define sem_name "/semaphore"
int main(int argc, char ** argv) {
    sem_t *sem;
    if ( argc != 2 ) {
        if ((sem = sem_open(sem_name, O_CREAT, 0777, 0)) == SEM_FAILED ) {
            fprintf(stderr, "sem_open error\n");
            return 1;
        }
        printf("veselkova.\nSemaphore is taken.\nWaiting for it to be dropped.\n");
        if (sem_wait(sem) < 0 )
            fprintf(stderr, "sem_wait error\n");
        if ( sem_close(sem) < 0 )
            fprintf(stderr, "sem_close error\n");
        return 0;
    }
    else {
        printf("Dropping semaphore...\n");
        if ( (sem = sem_open(sem_name, 0)) == SEM_FAILED ) {
            fprintf(stderr, "sem_open error\n");
            return 1;
        }
        sem_post(sem);
        printf("veselkova.\nSemaphore dropped.\n");
        return 0;
    }
}

[veselkova_anna@vpsj3IeQ ~]$ gcc -o sem sem.c -lpthread
[veselkova_anna@vpsj3IeQ ~]$ ./sem
veselkova.
Semaphore is taken.
Waiting for it to be dropped.
[veselkova_anna@vpsj3IeQ ~]$ ./sem 1
Dropping semaphore...
veselkova.
Semaphore dropped.
```

**Висновок:** Всі завдання були однаково складні.