

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 керуванням процесами в ОС Unix»**  
Варіант 6

Виконав:  
Студент групи АІ-203  
Сиваш А.І.

Одеса 2021

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

## 2 Завдання

### 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.

## Хід роботи

```
login as: sivash_andrij
sivash_andrij@91.219.60.189's password:
Last login: Mon May 17 00:24:41 2021 from 78.26.173.150
[sivash_andrij@vpsj3IeQ ~]$ mkfifo sivash
```

```
[sivash_andrij@vpsj3IeQ ~]$ ls -l | grep 'sivash'
-rw-rw-r-- 1 sivash_andrij sivash_andrij 1943597 Mar 21 19:42 \
```

Підключіть до іменованого каналу процес, який буде в нього писати

```
[sivash_andrij@vpsj3IeQ ~]$ chmod u=wr,g=,o= sivash
[sivash_andrij@vpsj3IeQ ~]$ ls /etc | grep '^g' > sivash
```

Створіть процес, який буде читати зі створеного раніше каналу

```
[sivash_andrij@vpsj3IeQ ~]$ cat sivash
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

```
[sivash_andrij@vpsj3IeQ ~]$ gzip -c < sivash > file1.gz
```

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

```
[sivash_andrij@vpsj3IeQ ~]$ cat sivash | gzip -c </etc/passwd> passwd_arch.gz
```

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

```

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

#define pipe_name "sivash"
#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 );
}

```

```

[sivash_andrij@vpsj3IeQ ~]$ gcc -o pipe channel.c
[sivash_andrij@vpsj3IeQ ~]$ ./pipe
sivash is created
sivash 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

```

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

[sivash_andrij@vpsj3IeQ ~]$ ls /etc | grep '^g' > sivash | grip -c <
sivash> archive_file.gz

[sivash_andrij@vpsj3IeQ ~]$ ./channel
sivash is created
sivash is opened
Incoming message (49): gconf
gcrypt
ghostscript
glvnd
gnupg
groff
group

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

[sivash_andrij@vpsj3IeQ ~]$ ls /etc | grep '^g' > sivash | grip -c <
/etc/passwd> passwd_arch.gz

[sivash_andrij@vpsj3IeQ ~]$ ./channel
sivash is created
sivash is opened
Incoming message (49): gconf
gcrypt
ghostscript
glvnd
gnupg
groff
group

Incoming message (49): group-
grub2.cfg
grub.d
gshadow-
gss
gssp
Incoming 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>

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("sivash (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("sivash (f2): %d\n", i);
        sleep(1);
    }
    pthread_exit(0);
}
```

```

[sivash_andrij@vpsj3IeQ ~]$ gcc -o pthread pthread.c -lpthread
[sivash_andrij@vpsj3IeQ ~]$ ./pthread
sivash (f2): 1
sivash (f1): 1
sivash (f2): 2
sivash (f1): 2
sivash (f2): 3
sivash (f1): 3
sivash (f1): 4
sivash (f2): 4
sivash (f1): 5
sivash (f2): 5
sivash (f1): 6
sivash (f2): 6
sivash (f2): 7
sivash (f1): 7
sivash (f2): 8
sivash (f1): 8
sivash (f2): 9
sivash (f1): 9

```

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

```

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

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

```



```
[sivash_andrij@vpsj3IeQ ~]$ nano semaphor.c
[sivash_andrij@vpsj3IeQ ~]$ gcc -o semaphor semaphor.c -lpthread
[sivash_andrij@vpsj3IeQ ~]$ ./semaphor
sem_open. sivash semaphore is taken.
Waiting for it to be dropped.
[sivash_andrij@vpsj3IeQ ~]$ ls /etc | grep "^k" > sivash
[sivash_andrij@vpsj3IeQ ~]$ ./semaphor 1
Dropping sivash semaphore...
sem_post. sivash semaphore dropped[sivash_andrij@vpsj3IeQ ~]$
```

### **Висновок:**

В результаті виконання лабораторної роботи, були вивчені особливості обміну інформацією між процесами за допомогою іменованих каналів, керування потоками, а також синхронізацію процесів через семафори та м'ютекси. Усі завдання були однакові по складності.