

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»  
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ  
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ

Лабораторна робота №10  
з дисципліни  
“Операційні системи”  
Тема  
**««Програмування керуванням процесами в ОС Unix»**  
Варіант 6

Виконав:  
Студент групи АІ-203  
Шуляк М.Р.

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.

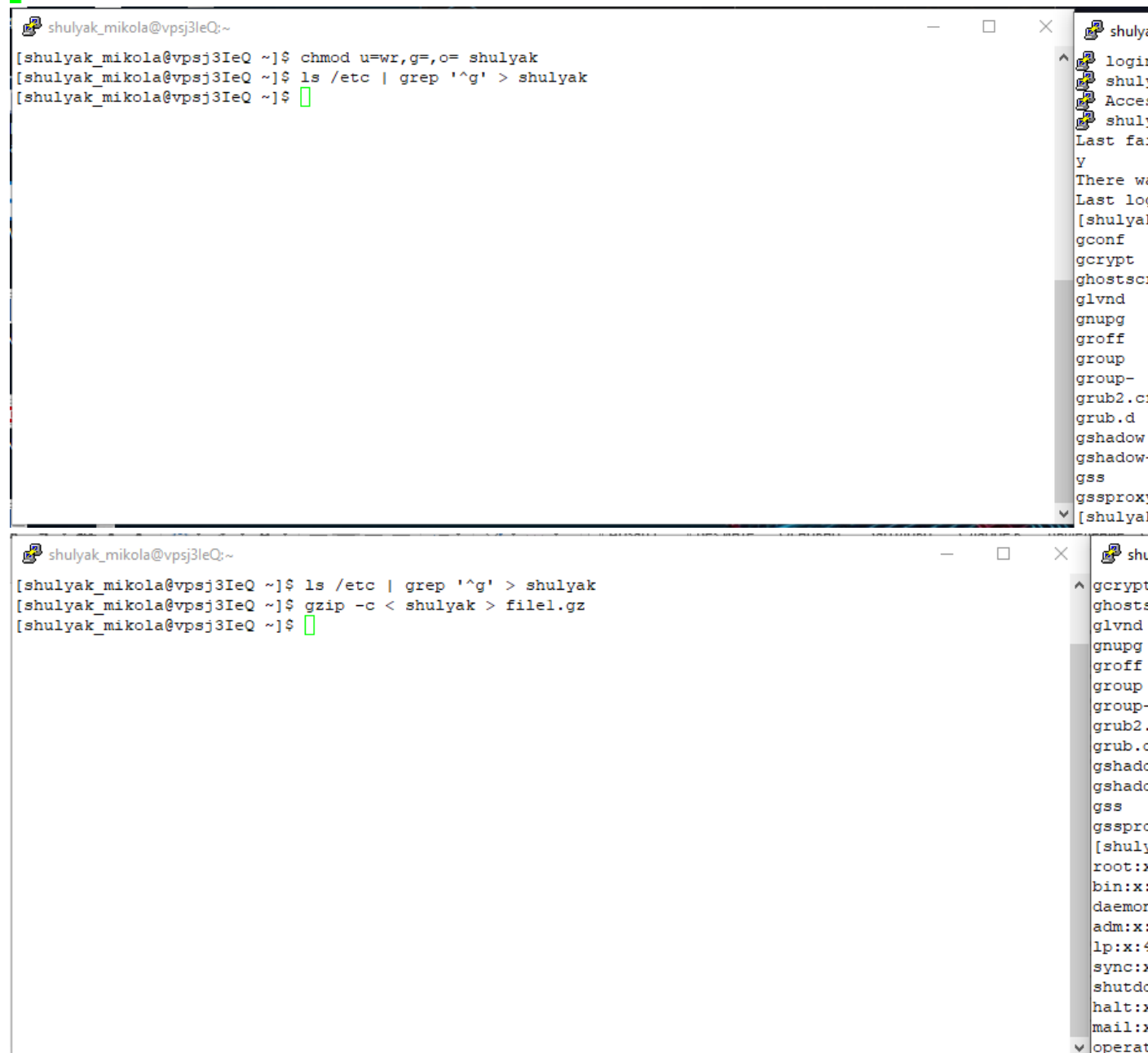
#### 2.4 Програмування семафорів

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

```
[shulyak_mikola@vpsj3IeQ ~]$ mkfifo shulyak
[shulyak_mikola@vpsj3IeQ ~]$ chmod u=rx,g=,o= shulyak
[shulyak_mikola@vpsj3IeQ ~]$ ls -l
>r-x----- 1 shulyak_mikola shulyak_mikola      0 Jun  2 20:48 shulyak
[shulyak_mikola@vpsj3IeQ ~]$ ls /etc | grep '^g' > shulyak
```



The image shows two screenshots of a terminal window. The first screenshot shows the user creating a file named 'shulyak' with permissions 'u=rx,g=,o=' and then listing the contents of the '/etc' directory, filtering for files starting with 'g'. The second screenshot shows the user listing the contents of the '/etc' directory, filtering for files starting with 'g', and then compressing the file 'shulyak' into 'file1.gz' using the 'gzip' command.

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

shulyak_mikola@vpsj3IeQ:~
[shulyak_mikola@vpsj3IeQ ~]$ ls /etc | grep '^g' > shulyak
[shulyak_mikola@vpsj3IeQ ~]$ gzip -c < shulyak > file1.gz
[shulyak_mikola@vpsj3IeQ ~]$
```

```

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


#define NAMEDPIPE_NAME "shulyak_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 mssg (%d): %s\n", len, buf);
    } while(1);
}

```

 shulyak\_mikola@vpsj3IeQ:~

```
[shulyak_mikola@vpsj3IeQ ~]$ ls /etc | grep '^g' > shulyak  
[shulyak_mikola@vpsj3IeQ ~]$ gzip -c < shulyak > file1.gz  
[shulyak_mikola@vpsj3IeQ ~]$ ls /etc | grep "g" > shulyak_2  
[shulyak_mikola@vpsj3IeQ ~]$ █
```

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

[ Wrote 34 lines ]

<b>^G</b> Get Help	<b>^O</b> WriteOut	<b>^R</b> Read File	<b>^Y</b> Prev Page	<b>^K</b> Cut Text	<b>^C</b> C
<b>^X</b> Exit	<b>^J</b> Justify	<b>^W</b> Where Is	<b>^V</b> Next Page	<b>^U</b> UnCut Text	<b>^T</b> T

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



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

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

[ Wrote 33 lines ]

**^G** Get Help  
**^X** Exit

**^O** WriteOut  
**^J** Justify

**^R** Read File  
**^W** Where Is

**^Y** Prev Page  
**^V** Next Page

**^K**  
**^U**

### 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.