

Міністерство освіти і науки України
Державний Університет Одеська Політехніка
Інститут комп'ютерних систем
Кафедра інформаційних систем

Лабораторна робота №12
з дисципліни «Операційні Системи»

«Програмування міжпроцесної та багатопоточної взаємодії»

Виконав:

ст. гр. АІ-204

Бериславський В.Р

Перевірив:

Блажко О. А.

Дрозд М.О.

Одеса – 2021

Мета:

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

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).

```
berislavskij_vladislav@vpsj3IeQ:~  
login as: berislavskij_vladislav  
berislavskij_vladislav@91.219.60.189's password:  
Last login: Mon May 17 06:41:19 2021 from 188.163.101.235  
[berislavskij_vladislav@vpsj3IeQ ~]$ mkfif berislavskij  
-bash: mkfif: command not found  
[berislavskij_vladislav@vpsj3IeQ ~]$ mkfifo berislavskij  
  
[berislavskij_vladislav@vpsj3IeQ ~]$ chmod u=rw,g=,o= berislavskij  
[berislavskij_vladislav@vpsj3IeQ ~]$ ls /etc | grep "^b" > berislavskij  
[berislavskij_vladislav@vpsj3IeQ ~]$
```

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.

```
berislavskij_vladislav@vpsj3IeQ:~  
login as: berislavskij_vladislav  
berislavskij_vladislav@91.219.60.189's password:  
Last login: Mon May 24 01:40:35 2021 from 188.163.100.178  
[berislavskij_vladislav@vpsj3IeQ ~]$ cat berislavskij  
bash_completion.d  
bashrc  
binfmt.d  
[berislavskij_vladislav@vpsj3IeQ ~]$
```

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`

```
[berislavskij_vladislav@vpsj3IeQ ~]$ gzip -c < berislavskij > file1.gz
```

```
berislavskij_vladislav@vpsj3IeQ:~  
[berislavskij_vladislav@vpsj3IeQ ~]$ cat /etc/passwd > berislavskij | 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  
saslauthd: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  
emps:x:996:1001:/:/home/emps:/bin/bash  
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin  
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin  
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin  
oracle:x:54321:54321:/:/home/oracle:/bin/bash  
apache:x:995:995:/:/home/apache:/sbin/nologin  
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/false  
named:x:25:25:Named:/var/named:/sbin/nologin  
exim:x:93:93:/:/var/spool/exim:/sbin/nologin  
vmmail:x:5000:5000:/:/var/local/vmail:/bin/bash
```

2.

```
berislavskij_vladislav@vpsj3IeQ:~  
GNU nano 2.3.1 File: imkanal.c  
  
#include <sys/stat.h>  
#include <fcntl.h>  
#include <string.h>  
#include <stdio.h>  
#define NAMEDPIPE_NAME "berislavskij_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!");  
        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("Incomming message (%d): %s\n", len, buf);  
    } while ( 1 );  
}
```

```
[berislavskij_vladislav@vpsj3IeQ ~]$ ls /etc | grep "^b" > berislavskij_2  
[berislavskij_vladislav@vpsj3IeQ ~]$  
  
[berislavskij_vladislav@vpsj3IeQ ~]$ nano imkanal.c  
[berislavskij_vladislav@vpsj3IeQ ~]$ gcc imkanal.c -o imkanal  
[berislavskij_vladislav@vpsj3IeQ ~]$ ./imkanal  
berislavskij_2 is created  
berislavskij_2 is opened  
Incomming message (34): bash_completion.d  
bashrc  
binfmt.d  
  
END! [berislavskij_vladislav@vpsj3IeQ ~]$
```

2.3 Програмування потоків

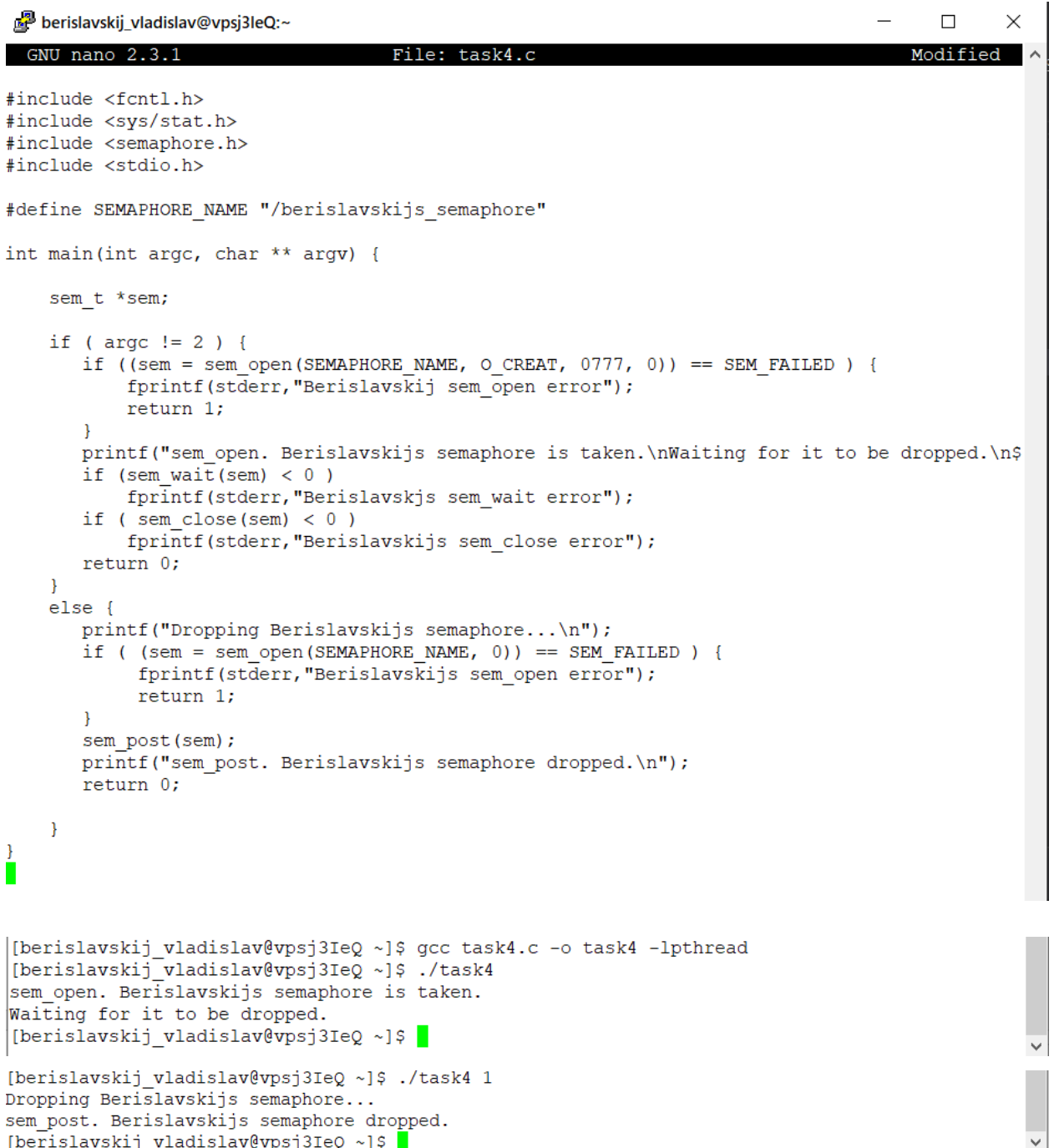
За прикладом з рисунку 2 розробіть програму керування потоками, в якій в повідомленнях буде вказано ваше прізвище латиницею. Виконайте програму за вказаним прикладом.

```
berislavskij_vladislav@vpsj3IeQ:~  
GNU nano 2.3.1 New Buffer Modified  
  
#include <pthread.h>  
#include <stdio.h>  
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);  
}  
void *f1(int *x) {  
    int i,n;  
    n = *x;  
    for (i=1;i<n;i++) {  
        printf("Berislavskij (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("Berislavskij (f2): %d\n", i);  
        sleep(1);  
    }  
    pthread_exit(0);  
}
```

```
[berislavskij_vladislav@vpsj3IeQ ~]$ gcc task3.c -o task3 -lpthread  
[berislavskij_vladislav@vpsj3IeQ ~]$ ./task3  
Berislavskij (f2): 1  
Berislavskij (f1): 1  
Berislavskij (f2): 2  
Berislavskij (f1): 2  
Berislavskij (f2): 3  
Berislavskij (f1): 3  
Berislavskij (f2): 4  
Berislavskij (f1): 4  
Berislavskij (f2): 5  
Berislavskij (f1): 5  
Berislavskij (f2): 6  
Berislavskij (f1): 6  
Berislavskij (f2): 7  
Berislavskij (f1): 7  
Berislavskij (f2): 8  
Berislavskij (f1): 8  
Berislavskij (f2): 9  
Berislavskij (f1): 9  
[berislavskij_vladislav@vpsj3IeQ ~]$
```

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.



```
berislavskij_vladislav@vpsj3IeQ:~  
GNU nano 2.3.1 File: task4.c Modified  
  
#include <fcntl.h>  
#include <sys/stat.h>  
#include <semaphore.h>  
#include <stdio.h>  
  
#define SEMAPHORE_NAME "/berislavskijs_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, "Berislavskij sem_open error");  
            return 1;  
        }  
        printf("sem_open. Berislavskijs semaphore is taken.\nWaiting for it to be dropped.\n");  
        if (sem_wait(sem) < 0 )  
            fprintf(stderr, "Berislavskijs sem_wait error");  
        if ( sem_close(sem) < 0 )  
            fprintf(stderr, "Berislavskijs sem_close error");  
        return 0;  
    }  
    else {  
        printf("Dropping Berislavskijs semaphore...\n");  
        if ( (sem = sem_open(SEMAPHORE_NAME, 0)) == SEM_FAILED ) {  
            fprintf(stderr, "Berislavskijs sem_open error");  
            return 1;  
        }  
        sem_post(sem);  
        printf("sem_post. Berislavskijs semaphore dropped.\n");  
        return 0;  
    }  
}  
[berislavskij_vladislav@vpsj3IeQ ~]$ gcc task4.c -o task4 -lpthread  
[berislavskij_vladislav@vpsj3IeQ ~]$ ./task4  
sem_open. Berislavskijs semaphore is taken.  
Waiting for it to be dropped.  
[berislavskij_vladislav@vpsj3IeQ ~]$  
  
[berislavskij_vladislav@vpsj3IeQ ~]$ ./task4 1  
Dropping Berislavskijs semaphore...  
sem_post. Berislavskijs semaphore dropped.  
[berislavskij_vladislav@vpsj3IeQ ~]$
```

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