

Міністерство освіти та науки України
Львівський національний університет ім. Івана Франка
Факультет електроніки та комп'ютерних технологій
Кафедра радіоелектронних і комп'ютерних систем

ЗВІТ

про виконання лабораторної роботи №7
«Реалізація міжпроцесової взаємодії на основі
інтерфейсу файлової системи»

Виконав:
Студент групи ФЕІ-23
Ангелов М.С.
Перевірив:
ас. Сінькевич О.О.

Львів – 2019

1. Завдання

- 1) Розробити систему обміну даними про поточну температуру повітря для Linux.
- 2) Розробити просту клієнт-серверну систему для Linux і Windows XP з використанням поіменованих каналів. Сервер повинен знаходити на диску файл і направляти його вміст клієнту, котрий після отримання цих даних має відобразити їх. Якщо файл не знайдено, сервер повертає рядок з повідомлення про помилку.

2. Виконання

2.1. Частина 1 (Температура)

- 1) Код клієнту

```
1  #include <sys/mman.h>
2  #include <stdio.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6
7  int main() {
8      printf("Client\n");
9      int fd = open("temperature.txt", O_RDWR | O_CREAT, 0644);
10     int* fmap = (int *)mmap(0, 100, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
11     close(fd);
12
13     for(;;) {
14         printf("%d\n", fmap[0]);
15         sleep(1);
16     }
17     munmap(fmap, 100);
18     return 0;
19 }
```

- 2) Код серверу

```
1  #include <sys/mman.h>
2  #include <stdio.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7
8  int main() {
9      printf("Server\n");
10     int fd = open("temperature.txt", O_RDWR | O_CREAT, 0644);
11     lseek(fd, 100, SEEK_SET);
12     write(fd, "", 100);
13     int* fmap = (int *)mmap(0, 100, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
14     close(fd);
15     struct flock lock = {0};
16     for(;;) {
17         lock.l_type = F_SETFL;
18         fcntl(fmap, F_SETLKW, &lock);
19         fmap[0] = rand() % 35;
20         lock.l_type = F_UNLCK;
21         fcntl(fmap, F_SETLKW, &lock);
22         sleep(10);
23     }
24     munmap(fmap, 100);
25     return 0;
26 }
```

3) Виконання з командного рядка.

```
nick@nick:~/lab7$ ./server Bus error (core dumped)
Server
nick@nick:~/lab7$ ./client
Client
8
8
8
8
8
8
8
8
8
8
8
11
11
11
```

2.2. Частина 2 (Використання каналів)

1) Код клієнту

```
1  #include <sys/types.h>
2  #include <sys/stat.h>
3  #include <fcntl.h>
4  #include <stdio.h>
5  #include <unistd.h>
6  #include <string.h>
7
8  int main() {
9      int fdl = open("buffer.txt", O_RDWR);
10     char buf [100];
11     char file_buf[100];
12     while(1) {
13         printf(">> ");
14         scanf("%s", &buf);
15         write(fdl, buf, sizeof(buf));
16         if(strcmp("exit", buf) == 0) {
17             break;
18         } else {
19             read(fdl, file_buf, sizeof(file_buf));
20             printf("%s\n", file_buf);
21         }
22     }
23     close(fdl);
24 }
```

2) Код серверу

```
1 #include <sys/stat.h>
2 #include <fcntl.h>
3 #include <stdio.h>
4 #include <unistd.h>
5 #include <string.h>
6
7 int main() {
8     int fd, bytes_read; char read_buf[100];
9     char file_buf[100];
10    mkfifo("buffer.txt", 0644);
11    for (;;) {
12        fd = open("buffer.txt", O_RDWR);
13        bytes_read = read(fd, read_buf, sizeof(read_buf));
14
15        if (bytes_read > 0) {
16            read_buf[bytes_read] = '\0';
17            printf("Got message: %s\n", read_buf);
18            if (strcmp(read_buf, "exit") == 0) {
19                close(fd);
20                return 0;
21            } else {
22                int c;
23                FILE *file;
24                file = fopen(read_buf, "r");
25                if (file) {
26                    while (fgets(file_buf, sizeof(file_buf), file) != NULL) {
27                        printf("%s\n", file_buf);
28                        write(fd, file_buf, sizeof(file_buf));
29                    }
30                    printf("\n");
31                    fclose(file);
32                } else {
33                    printf("File not found\n");
34                }
35            }
36        }
37        close(fd);
38    }
39}
```

3) Виконання з командного рядка.

```
nick@nick:~/lab7$ sudo ./server    nick@nick:~/lab7$ sudo ./client
Got message: exit                  >> exit
nick@nick:~/lab7$                 nick@nick:~/lab7$
```

3. Висновок

Під час виконання лабораторної роботи я дізнався про способи обміну даними між різними процесами у Linux та створив базову програму для демонстрації.