

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

**Дисципліна «Операційні системи»**

**Лабораторна робота “Канали”**

**Запоріжжя**

**2025**

### Завдання:

- Ознайомитись з системними викликами для роботи з каналами, такими як: **pipe()**, **mkfifo()**.
- Продемонструвати у роботі кожну з цих функцій (досить продемонструвати частину коду програми, де використовується функція).

Номер індивідуального завдання обчислюється за формулою:

$$(< \text{номер завдання} >) = (< \text{свій номер у списку} > - 1) \% 3 + 1$$

Свій номер у списку дивись на останній сторінці поточного документу (Додаток 1).

- (1) Створити програму копіювання файлів (**fcopy**).
- (2) Створити програму пошуку у файлі заданої послідовності символів (**fsearch1**).
- (3) Створити програму пошуку у файлі заданої послідовності символів (**fsearch2**).

#### (1) Технічні умови (до програми **fcopy**):

- Читання та запис файлу відбувається у паралельному режимі.
- Програма копіює будь які файли (текстові або бінарні) будь якого розміру.
- Ім'я файлів (оригінала та копії) вказуються у командному рядку.
- Максимальний обсяг даних, який може бути завантажений у пам'ять не повинен перевищувати **64 кБ**
- Впродовж копіювання файлу на екран виводиться прогрес виконаної роботи. Формат прогресу вибирається самостійно.
- По завершенню копіювання програма повідомляє скільки байтів було скопійовано та впродовж якого часу (у секундах).

#### (2) Технічні умови (до програми **fsearch1**):

- Файл відкриває батьківський процес та читає його, пошук відбувається у дочірньому процесі.
- Програма обробляє тільки текстові файли.
- Ім'я файлу та послідовність символів до пошуку вказуються у командному рядку. Послідовність символів **ASCII** для пошуку записується у лапках.
- Максимальний обсяг даних, який може бути завантажений у пам'ять не повинен перевищувати **64 кБ**
- Впродовж пошуку на екран виводиться прогрес виконаної роботи. Формат прогресу вибирається самостійно.

- По завершенню пошуку програма повідомляє результат пошуку (знайдено чи ні), скільки байтів було переглянуто (до моменту знаходження вказаної послідовності символів) та впродовж якого часу (у секундах). Якщо вказана послідовність символів знайдена, то повідомляється індекс входження послідовності символів у файлі (номер байту з якого починається знайдена послідовність), у протилежному випадку повідомляється про її відсутність.

### (3) Технічні умови (до програми **fsearch2**):

- Файл відкриває дочірний процес та читає його, пошук відбувається у батьківському процесі.
- Програма обробляє тільки текстові файли.
- Ім'я файлу та послідовність символів до пошуку вказуються у командному рядку. Послідовність символів **ASCII** для пошуку записується у лапках.
- Максимальний обсяг даних, який може бути завантажений у пам'ять не повинен перевищувати **64 кБ**
- Впродовж пошуку на екран виводиться прогрес виконаної роботи. Формат прогресу вибирається самостійно.
- По завершенню пошуку програма повідомляє результат пошуку (знайдено чи ні), скільки байтів було переглянуто (до моменту знаходження вказаної послідовності символів) та впродовж якого часу (у секундах). Якщо вказана послідовність символів знайдена, то повідомляється індекс входження послідовності символів у файлі (номер байту з якого починається знайдена послідовність), у протилежному випадку повідомляється про її відсутність.

#### **Звіт:**

Звіт оформлюється відповідно стандарту. Обов'язкові розділи звіту:

- тема роботи
- цілі роботи
- завдання
- інструменти (мова програмування, параметри комп'ютера на якому тестувалися програми)
- теоритична частина
- практична частина

- загальний опис програми
- розбиття програми на блоки, блок-схема програми
- текст програми з коментарем
- протокол роботи програми для кожного з тестів
- висновки

## Теоритичні відомості

Канали – це один з простих механізмів між процесної взаємодії. Існує два типу каналів: **анонімний** та **іменований**. Перший застосовується між процесами, які пов’язані родинними стосунками (за звичай між батьківським та дочиннім процесами), другий – між будь якими процесами.

Типи каналів:

### **анонімний**

створюються за допомогою системного виклику **pipe()**

працюють тільки між процесами з родинними відносинами

### **іменований**

створюються за допомогою бібліотечної функції **mkfifo()**

працюють між будь якими процесами

існують у файловій системі

Для запису даних у канал та їх читання використовують вже знайомі системні виклики **write()** та **read()**.

### **Анонімний канал pipe()**

```
#include <unistd.h>
```

```
int pipe(int pipefd[2]);
```

- **pipefd[0]** – дескриптор для читання
- **pipefd[1]** – дескриптор для запису

### **Приклад:**

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

```
#include <stdlib.h>
```

```
#include <unistd.h>
```

```
#include <sys/wait.h>
```

```
int main() {
```

```
    int pipefd[2];
```

```
    pipe(pipefd);           // Створюємо канал
```

```
    pid_t pid = fork();    // Створюємо процес
```

```
    char buffer[4096];     // Створюємо буфер
```

```
    if (pid == 0) {
```

```

// Дочірній процес
close(pipefd[1]);          // Закриваємо дескриптор для запису
size_t size = read(pipefd[0], buffer, sizeof(buffer));
printf("Child read: %s\n", buffer);
close(pipefd[0]);
exit(0);
}

// Батьківський процес
close(pipefd[0]);          // Закриваємо дескриптор для читання
char *msg = "Hello from parent!\n";
size_t size = write(pipefd[1], msg, sizeof(msg));
close(pipefd[1]);

wait(NULL);

return 0;
}

```

Батьківський процес створює канал, отримує два дескриптори. Оскільки цей процес повинен записувати дані у канал, то дескриптор для читання у батьківському процесі закриваємо. Потім створюється дочірній процес (який є копією батьківського процесу) у якому будемо тільки читати дані з каналу. Тому дескриптор для запису у дочірньому процесі закриваємо.

### Іменованний канал

```

#include <sys/types.h>
#include <sys/stat.h>
int mkfifo(const char* name, int flags);

```

### Приклад:

#### Письменник:

```

#include <unistd.h>
#include <stdlib.h>
#include <fcntl.h>

#include <sys/types.h>
#include <sys/stat.h>

int main() {

```

```
if (mkfifo("/tmp/yun.dat", 0600) < 0) {
    write(2, "Error mkfifo\n", 13);
    exit(0);
}
int fd = open("/tmp/yun.dat", O_WRONLY);
size_t size = write(fd, "Hello!\n", 7);
close(fd);
unlink("/tmp/yun.dat");           // Видалити канал
exit(0);
}
```

Читач:

```
#include <unistd.h>
#include <stdlib.h>
#include <fcntl.h>

#include <sys/types.h>
#include <sys/stat.h>

#define SIZE_BUF 4096
char buf[SIZE_BUF];

int main() {
    int fd = open("/tmp/yun.dat", O_RDONLY);
    if (fd < 0) {
        write(2, "Error open\n", 11);
        close(1);
    }
    size_t size = read(fd, buf, SIZE_BUF);
    write(1, buf, size);
    close(fd);
    exit(0);
}
```

**Додаток 1**

Список студентів по курсу «Операційні системи» для визнання номеру індивідуального завдання.

<b>1</b>	Шевченко Богдан Миколайович	ylua5555@gmail.com
<b>2</b>	Адамов Ярослав Родіонович	larchistrike@gmail.com
<b>3</b>	Астапенко Кирило Тарасович	kirill.ast98@gmail.com
<b>4</b>	Башлай Владислав Віталійович	vladbashlay123@gmail.com
<b>5</b>	Бурмагін Нікіта Сергійович	nnburmagin@gmail.com
<b>6</b>	Дереча Олексій Іванович	lollexa215@gmail.com
<b>7</b>	Доценко Денис Олексійович	denis.dotsenko.al@gmail.com
<b>8</b>	Дячко Діана Євгенівна	dianadiachko@gmail.com
<b>9</b>	Євстігнєєв Ростислав Денисович	yevstihnieiev.uni@gmail.com
<b>10</b>	Згурський Данило Олександрович	danilzgurs@gmail.com
<b>11</b>	Здор Віталій Віталійович	azot587@gmail.com
<b>12</b>	Іванченко Софія Олександрівна	sonyaivanchenko1@gmail.com
<b>13</b>	Кабанов Артем Іванович	artemkabanov108@gmail.com
<b>14</b>	Каптюх Костянтин Денисович	ravolut1onr@gmail.com
<b>15</b>	Караваєв Олександр Олександрович	koshak327@gmail.com
<b>16</b>	Колеснік Максим Олександрович	maksimkolesnik99@gmail.com
<b>17</b>	Крутько Дмитро Геннадійович	durkomom735@gmail.com
<b>18</b>	Кузін Іван Олексійович	ivankuzin3049@gmail.com
<b>19</b>	Лукіна Дарія Дмитрівна	lukinadariia@gmail.com
<b>20</b>	Матяш Ярослав Олександрович	matyhya23@gmail.com
<b>21</b>	Мірошніченко Богдан Володимирович	bogdanmirosnicenko@gmail.com
<b>22</b>	Могилін Владислав Олександрович	mogilinvlad@gmail.com
<b>23</b>	Мороко Ярослав Владиславович	legionemptys@gmail.com
<b>24</b>	Петрик Денис Сергійович	denispetrik714@gmail.com
<b>25</b>	Пригарін Богдан Сергійович	pryharin.bohdan@gmail.com
<b>26</b>	Сандак Поліна Андріївна	charasailar@gmail.com
<b>27</b>	Сапа Олександр Олександрович	sahasapa2000@gmail.com
<b>28</b>	Сапа Сергій Олександрович	serchie228@gmail.com
<b>29</b>	Тарасенко Нікіта Сергійович	tarasenkonikita27@gmail.com
<b>30</b>	Темченко Станіслав Андрійович	temchenkosv8@gmail.com
<b>31</b>	Хоренженко Іван Сергійович	ivankhorenzhenko22@gmail.com
<b>32</b>	Черевичний Ярослав Сергійович	yarik16122006@gmail.com
<b>33</b>	Черненко Едуард Кирилович	edikchernenko133@gmail.com
<b>34</b>	Шарій Юрій Андрійович	shariyya777@gmail.com