

Лабораторная работа №14

Дисциплина: Операционные системы

Маслова Анастасия Сергеевна

Содержание

Цель работы

Приобретение практических навыков работы с именованными каналами.

Задание

Изучите приведённые в тексте программы `server.c` и `client.c`. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения: 1. Работает не 1 клиент, а несколько (например, два). 2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию `sleep()` для приостановки работы клиента. 3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию `clock()` для определения времени работы сервера. Что будет в случае, если сервер завершит работу, не закрыв канал?

Теоретическое введение

Одним из видов взаимодействия между процессами в операционных системах является обмен сообщениями. Под сообщением понимается последовательность байтов, передаваемая от одного процесса другому. В операционных системах типа UNIX есть 3 вида межпроцессорных взаимодействий: общедоступные (именованные каналы, сигналы), System V Interface Definition (SVID — разделяемая память, очередь сообщений, семафоры) и BSD (сокеты). Для передачи данных между неродственными процессами можно использовать механизм именованных каналов (named pipes). Данные передаются по принципу FIFO (First In First Out) (первым записан — первым прочитан), поэтому они называются также FIFO pipes или просто FIFO. Именованные каналы отличаются от неименованных наличием идентификатора канала, который представлен как специальный файл (соответственно имя именованного канала — это имя файла). Поскольку файл находится на локальной файловой системе, данное IPC используется внутри одной системы.

Более подробно - [здесь](#).

Выполнение лабораторной работы

1. Работает не 1 клиент, а несколько (например, два).

```
[asmaslova@fedora 1]$ ./server
FIFO Server...
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
[asmaslova@fedora 1]$
```

```
[asmaslova@fedora 1]$ ./client1
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
[asmaslova@fedora 1]$
```

```
#include "common.h"
```

```
int main()
{
    int readfd; /* дескриптор для чтения из FIFO */
    int n;
    char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */

    printf("FIFO Server...\n");

    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
    {
        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-1);
    }

    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
    {
        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-2);
    }

    while((n = read(readfd, buff, MAX_BUFF)) > 0)
    {
        if(write(1, buff, n) != n)
        {
            fprintf(stderr, "%s: Ошибка вывода (%s)\n",
                __FILE__, strerror(errno));
            exit(-3);
        }
    }

    close(readfd);

    if(unlink(FIFO_NAME) < 0)
    {
        fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
            __FILE__, strerror(errno));
        exit(-4);
    }

    exit(0);
}
```

```
U:--- server.c All L1 (C/*l Abbrev)
```

```
U:--- client.c      All L1      (C/*l Abbrev)
```

- ```
[asmaslova@fedora 2]$./server
FIFO Server...
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!

server timeout
30 seconds passed!

```

```
[asmaslova@fedora 2]$./client
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
[asmaslova@fedora 2]$
```

```

#include "common.h"

int main()
{
 int readfd; /* дескриптор для чтения из FIFO */
 int n;
 char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */

 printf("FIFO Server...\n");

 if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
 {
 fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
 __FILE__, strerror(errno));
 exit(-1);
 }

 if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
 {
 fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
 __FILE__, strerror(errno));
 exit(-2);
 }

 clock_t now=time(NULL), start=time(NULL);
 while(now-start<30)
 {
 while((n = read(readfd, buff, MAX_BUFF)) > 0)
 {
 if(write(1, buff, n) != n)
 {
 fprintf(stderr, "%s: Ошибка вывода (%s)\n",
 __FILE__, strerror(errno));
 exit(-3);
 }
 }
 now=time(NULL);
 }

 printf("\n-----\nserver timeout\n%i seconds passed!\n-----\n", now-start);
 close(readfd);

 if(unlink(FIFO_NAME) < 0)
 {
 fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
 __FILE__, strerror(errno));
 }
}

```

U:--- **server.c** Top L16 (C/\*l Abbrev)

Beginning of buffer

```

clock_t now=time(NULL), start=time(NULL);
while(now-start<30)
{
 while((n = read(readfd, buff, MAX_BUFF)) > 0)
 {
 if(write(1, buff, n) != n)
 {
 fprintf(stderr, "%s: Ошибка вывода (%s)\n",
 __FILE__, strerror(errno));
 exit(-3);
 }
 }
 now=time(NULL);
}

printf("\n-----\nserver timeout\n%i seconds passed!\n-----\n", now-start);
close(readfd);

if(unlink(FIFO_NAME) < 0)
{
 fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
 __FILE__, strerror(errno));
 exit(-4);
}

exit(0);
}

```

U:--- **server.c** Bot L26 (C/\*l Abbrev)

```

#include "common.h"

#define MESSAGE "Hello Server!!!\n"

int main()
{
 int writefd; /* дескриптор для записи в FIFO */
 int msglen;
 char message[10];
 int count;
 long long int T;
 for(count=0; count<=5; ++count)
 {
 sleep(5);
 T=(long long int) time(0);
 sprintf(message,"%lli", T);
 message[9] = '\n';
 printf("FIFO Client...\n");

 if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
 {
 fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
 __FILE__, strerror(errno));
 exit(-1);
 }

 msglen = strlen(MESSAGE);
 if(write(writefd, MESSAGE, msglen) != msglen)
 {
 fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
 __FILE__, strerror(errno));
 exit(-2);
 }
 }
 close(writefd);
 exit(0);
}

```

```

U:--- client.c All L1 (C/*l Abbrev)
Beginning of buffer

```

- Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию clock() для определения времени работы сервера. Что будет в случае, если сервер завершит работу, не закрыв

```

[asmaslova@fedora 3]$./server
FIFO Server...
Hello Server!!!

server timeout
30 seconds passed!

[asmaslova@fedora 3]$./client
FIFO Client...

```

канал?

```
File Edit Options Buffers Tools C Help
[Icons: New, Open, Save, Close, Save All, Undo, Redo, Copy, Paste, Find]

#include "common.h"

int main()
{
 int readfd; /* дескриптор для чтения из FIFO */
 int n;
 char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */

 printf("FIFO Server...\n");

 if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
 {
 fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n",
 __FILE__, strerror(errno));
 exit(-1);
 }

 if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
 {
 fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
 __FILE__, strerror(errno));
 exit(-2);
 }

 clock_t now=time(NULL), start=time(NULL);
 while(now-start<30)
 {
 while((n = read(readfd, buff, MAX_BUFF)) > 0)
 {
 if(write(1, buff, n) != n)
 {
 fprintf(stderr, "%s: Ошибка вывода (%s)\n",
 __FILE__, strerror(errno));
 exit(-3);
 }
 }
 now=time(NULL);
 }

 printf("\n-----\nServer timeout\n%li seconds passed!\n-----\n", now-start);
 close(readfd);

 if(unlink(FIFO_NAME) < 0)
 {
 fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
 __FILE__, strerror(errno));
 }
}
```

U:\*\*- **server.c** Top L40 (C/\*l Abbrev)

Beginning of buffer



```

clock_t now=time(NULL), start=time(NULL);
while(now-start<30)
{
 while((n = read(readfd, buff, MAX_BUFF)) > 0)
 {
 if(write(1, buff, n) != n)
 {
 fprintf(stderr, "%s: Ошибка вывода (%s)\n",
 __FILE__, strerror(errno));
 exit(-3);
 }
 }
 now=time(NULL);
}

printf("\n-----\nServer timeout\n%i seconds passed!\n-----\n", now-start);
close(readfd);

if(unlink(FIFO_NAME) < 0)
{
 fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n",
 __FILE__, strerror(errno));
 exit(-4);
}

exit(0);
}

```

U:\*\*\* **server.c** Bot L40 (C/\*l Abbrev)

```

#include "common.h"

#define MESSAGE "Hello Server!!!\n"

int main()
{
 int writefd; /* дескриптор для записи в FIFO */
 int msglen;

 printf("FIFO Client...\n");

 if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
 {
 fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
 __FILE__, strerror(errno));
 exit(-1);
 }

 msglen = strlen(MESSAGE);
 if(write(writefd, MESSAGE, msglen) != msglen)
 {
 fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
 __FILE__, strerror(errno));
 exit(-2);
 }

 close(writefd);

 exit(0);
}

```

```

U:--- client.c All L1 (C/*l Abbrev)
Beginning of buffer

```

## Выводы

В ходе выполнения лабораторной работы я приобрела практические навыки работы с именованными каналами.

## Список литературы