Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №6
за 4 семестр
По дисциплине: «ОСиСП»
Тема: «Средства межпроцессного взаимодействия»

Выполнил:
Студент 2 курса
Группы ПО-6(1)
Мартынович Д. М.
Проверил:
Давидюк Ю. И.

2022

Лабораторная работа №6

Средства межпроцессного взаимодействия

Вариант 8

Цель работы: изучить работу с средствами межпроцессного взаимодействия в ОС Linux.

Задание для выполнения

Ознакомиться с руководством, теоретическими сведениями и лекционным материалом по использованию и функционированию средств взаимодействия.

Написать программу, которая порождает дочерний процесс, и общается с ним через средства взаимодействия согласно варианту (табл.А), передавая и получая информацию согласно варианту (табл.Б). Передачу и получение информации каждым из процессов сопровождать выводом на экран информации типа "процесс такой-то передал/получил такую-то информацию". **Дочерние процессы начинают операции после получения сигнала SIGUSR1 от родительского процесса.**

8	Очереди сообщений	Родитель передает три строки, потомок возвращает самую длинную из них.

Код программы :

```
#include <stdlib.h>
#include <signal.h>
#include <string.h>
#include <errno.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <mqueue.h>
void sw_processor() {
      printf("Processor work\n");
}
int main(){
    mqd_t mqID;
    char str1[100];
    char str2[100];
    char str3[100];
    int proces;
    printf("Input 3 strok:\n ");
    scanf("%s",str1);
    scanf("%s",str2);
```

```
scanf("%s",str3);
printf("\n");
(void)signal(SIGUSR1, sw_processor);
//Создание очереди сообщений
mqID=mq_open("/Queue1",O_RDWR | O_CREAT | O_EXCL, 0666,NULL);
if (mqID<0)
{
    if(errno==EEXIST)
    {
        mq_unlink("/Queue1");
        mqID = mq_open("/Queue1",O_RDWR | O_CREAT, 0666, NULL);
    }
    else
    {
        perror("open message queue error...");
        return -1;
    }
}
proces=fork();
if (proces<0)
{
    printf("Can't fork child process\n");
    return -1;
}
else if (proces>0)
{
    //parent
    printf("Parent (pid=%d)\n",getpid());
    //Отправка в очередь
    if(mq_send(mqID,str1,sizeof(str1),1)<0)
    {
    printf("send str1 failed \n");
    }
    printf("Process (pid=%d) Send str1 succesfull \n",getpid());
    if(mq_send(mqID,str2,sizeof(str2),1)<0)
    {
    printf("send str2 failed \n");
    }
    printf("Process (pid=%d) Send str2 succesfull \n",getpid());
    if(mq_send(mqID,str3,sizeof(str3),1)<0)
    {
    printf("send str1 failed \n");
    }
    printf("Process (pid=%d) Send str3 succesfull \n",getpid());
    kill(proces,SIGUSR1);
    sleep(3);

    struct mq_attr mqAttr;
    mq_getattr(mqID,&mqAttr);
```

```
        char longestSTR[mqAttr.mq_msgsize];
        if (mq_receive(mqID,longestSTR,mqAttr.mq_msgsize,NULL)<0)
        {
            printf("receive message failed...\n");
            perror("error info");
        }
        printf("Procces parent with (pid=%d) receive message 4\n",getpid());
        printf("The longest word is ""%s"" with %d symbols\
n",longestSTR,strlen(longestSTR));
        printf("Parent (pid=%d) exit\n", getpid());
    }
    else
    {
        //child
        printf("Child (pid=%d),connected to (ppid=%d)\n",getpid(),getppid());
        struct mq_attr mqAttr;
        mq_getattr(mqID,&mqAttr);
        char
gStr1[mqAttr.mq_msgsize],gStr3[mqAttr.mq_msgsize],gStr2[mqAttr.mq_msgsize];
        char maxStr[100];
        int maxtemp=0;
        //получение очереди сообщений
        if (mq_receive(mqID,gStr1,mqAttr.mq_msgsize,NULL)<0)
        {
            printf("receive message failed...\n");
            perror("error info");
        }
        printf("Procces (pid=%d) receive message 1: %s\n",getpid(),gStr1);
        if (mq_receive(mqID,gStr2,mqAttr.mq_msgsize,NULL)<0)
        {
            printf("receive message failed...\n");
            perror("error info");
        }
        printf("Procces (pid=%d) receive message 2: %s\n",getpid(),gStr2);
        if (mq_receive(mqID,gStr3,mqAttr.mq_msgsize,NULL)<0)
        {
            printf("receive message failed...\n");
            perror("error info");
        }
        printf("Procces (pid=%d) receive message 3: %s\n",getpid(),gStr3);

        if(maxtemp<strlen(gStr1))
        {
            maxtemp=strlen(gStr1);
            strcpy(maxStr, gStr1);
        }
        if(maxtemp<strlen(gStr2))
        {
            maxtemp=strlen(gStr1);
```

```
            strcpy(maxStr, gStr2);
        }
        if(maxtemp<strlen(gStr3))
        {
            maxtemp=strlen(gStr1);
            strcpy(maxStr, gStr3);
        }
        printf("\n");
        if(mq_send(mqID,maxStr,sizeof(maxStr),1)<0)
        {
        printf("send maxStr failed \n");
        }
        printf( "Child with (pid=%d) send maxStr succesfull \n",getpid());

        kill(proces,SIGUSR1);
        printf("Child (pid=%d) exit\n", getpid());
    }
    return 0;
}
```

```
ikrut0nardo@kali: ~/Desktop/lab6
└─$ ./main.out
Input 3 strok:
 qwer
qwe
q

Parent (pid=9559)
Process (pid=9559) Send str1 succesfull
Process (pid=9559) Send str2 succesfull
Process (pid=9559) Send str3 succesfull
Processor work
Child (pid=9561),connected to (ppid=9559)
Procces (pid=9561) receive message 1: qwer
Procces (pid=9561) receive message 2: qwe
Procces (pid=9561) receive message 3: q

Child with (pid=9561) send maxStr succesfull
Processor work
Child (pid=9561) exit
Processor work
Procces parent with (pid=9559) receive message 4
The longest word is qwer with 4 symbols
Parent (pid=9559) exit
```

Вывод: изучил работу с средствами межпроцессного взаимодействия в ОС Linux.