****Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работ №2 по курсу**

**«Операционные системы»**

Группа: М80 – 206Б-18

Студент: Касимов М.М.

Преподаватель: Соколов А.А.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2019.

**Содержание**

1. Постановка задачи
2. Общие сведения о программе
3. Общий метод и алгоритм решения
4. Основные файлы программы
5. Демонстрация работы программы
6. Вывод

**Постановка задачи**.

Рекурсивное вычисление факториала, где каждый отдельный уровень рекурсии

вычисляется в отдельном процессе.

**Общие сведения о программе**

Программа компилируется из одного файла lab2.c. В данном файле используются заголовочные файлы stdio.h, unistd.h, stdbool.h, stdlib.h, wait.h, sys/types.h . В программе используются следующие системные вызовы:

1. **read** – для чтения данных из файла
2. **write** – для записи данных в файл
3. **pipe** – для создания однонаправленного канала, через который могут общаться два процесса. Системный вызов возвращает два дескриптора файлов. Один для чтения из канала, другой для записи в канал.
4. **fork** – для создания дочернего процесса.
5. **wait** – для ожидания завершения дочернего процесса.

**Общий метод и алгоритм решения**.

Для реализации поставленной задачи необходимо:

1. Используя системный вызов pipe создать канал, по которому будут обмениваться данными два процесса.
2. Используя системный вызов fork создать дочерний процесс.
3. В родительском процессе считывать данные cо стандартного потока и правильно распарсить в целое число.
4. Как в родительском процессе данные считались, необходимо записать их в канал с помощью системного вызова write.
5. Пока родительский процесс на записал данные в канал. Дочерний процесс ждет. И как только родительский процесс записал данные в канал дочерний процесс считывает их, производит вычисления и возвращает результат родительскому процессу.
6. Родительский процесс выводит результат используя write.

**Основные файлы программы**.

**Файл main.c**

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <wait.h>

#include <stdlib.h>

#include <stdbool.h>

#include <stdlib.h>

unsigned long long fact(int n){

if (n == 0){

return 1;

}

else {

int write\_fd[2];

if (pipe(write\_fd)){

printf("Error: pipe\n");

exit(1);

}

pid\_t proc = fork();

if (proc < 0){

printf("Error: fork\n");

exit(1);

}

int wt;

wait(&wt);

if (proc == 0){

unsigned long long res;

res = fact(n - 1);

unsigned long long count;

count = write(write\_fd[1], &res, sizeof(int));

exit(0);

}

if (proc > 0) {

unsigned long long res;

unsigned long long count;

count = read(write\_fd[0], &res, sizeof(int));

return n \* (res);

}

}

}

int main(){

char a[132] = { 'I','n','s','t','r','u','c','t','i','o','n','.','\n',

'E','n','t','e','r',' ','o','n','l','y',' ','o','n','e',' ','n','o','n','n','e','g','a','t','i','v','e',' ',

'i','n','t','e','g','e','r',' ','n','u','m','b','e','r',' ','l','e','s','s',' ','t','h','a','n',

' ','1','4','.',' '

,'I','n',' ','c','a','s','e',' ','o','f',' ','i','n','c','o','r','r','e','c','t',' ','i','n','p','u','t',',',' ','t','h','e',' ','p','r','o'

,'g','r','a','m',' ','w','i','l','l',' ','s','i','m','p','l','y',' ','t','e','r','m','i','n','a','t','e',':','\n' };

for ( int i = 0 ; i < 132 ; ++i ) {

write(STDOUT\_FILENO,&a[i],sizeof(char));

}

int flag = 0,flagPlus = 0,flagTabs = 0,flagNumber = 0;

int n = 0;

char c;

while(true) {

read(STDIN\_FILENO,&c,1);

if (c <= '9' && c >= '0') {

flagPlus++;

flagNumber++;

n \*= 10;

n += c - '0';

continue;

}

if (c=='\n')

break;

if (c == '+' && flagPlus == 0) {

flagPlus++;

continue;

}

if ((c == ' ' || c == '\t') && (flagTabs == 0)) {

continue;

}

else

++flag;

}

if (flag != 0 || flagNumber == 0) {

return 0;

}

unsigned long long k;

k=fact(n);

char res[17],res2[17];

int i=0;

while (k!=0){

res[i]=k%10;

k/=10;

++i;}

char tmp;

for (int j = 0;j < i;++j){

res2[j]=res[i-j-1];}

for (int j = 0;j < i;++j){

tmp = res2[j]+'0';

write(STDOUT\_FILENO,&tmp,sizeof(char));

}

char enter = '\n';

write(STDOUT\_FILENO,&enter,sizeof(char));

return 0;

}

**Демонстрация работы программы.**

magomed@magomed-pc ~/OS/lab2/os/l2 $ ./a.out

Instruction.

Enter only one nonnegative integer number less than 14. In case of incorrect input, the program will simply terminate:

-2

magomed@magomed-pc ~/OS/lab2/os/l2 $ ./a.out

Instruction.

Enter only one nonnegative integer number less than 14. In case of incorrect input, the program will simply terminate:

13

6227020800

magomed@magomed-pc ~/OS/lab2/os/l2 $ ./a.out

Instruction.

Enter only one nonnegative integer number less than 14. In case of incorrect input, the program will simply terminate:

0

1

magomed@magomed-pc ~/OS/lab2/os/l2 $ ./a.out

Instruction.

Enter only one nonnegative integer number less than 14. In case of incorrect input, the program will simply terminate:

1n

magomed@magomed-pc ~/OS/lab2/os/l2 $

**Вывод.**

Я научился создавать процессы, используя системный вызов fork(), обрел навыки межпроцессного взаимодействия посредством каналов, которые создаются вызовом pipe(). Узнал что такое вектор прерываний и как работают процессы в ОС Linux. Улучшил навыки программирования на языке программирования Си.