**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Операционные системы»

**Курсовой проект**

Тема: Управление процессами в ОС

Студент: Калугин К. А.

Группа: М8О-207Б-19

Преподаватель: Миронов Е.С.

Дата:

Оценка:

Москва, 2021

**Постановка задачи:** Необходимо написать 3-и программы. Далее будем обозначать эти программы A, B, C. Программа A принимает из стандартного потока ввода строки, а далее их отправляет программе С. Отправка строк должна производится построчно. Программа C печатает в стандартый вывод, полученную строку от программы A. После получения программа C отправляет программе А сообщение о том, что строка получена. До тех пор пока программа А не примет «сообщение о получение строки» от программы С, она не может отправялять следующую строку программе С. Программа B пишет в стандартный вывод количество отправленных символов программой А и количество принятых символов программой С. Данную информацию программа B получает от программ A и C соответственно. Способ организация межпроцессорного взаимодействия выбирает студент.

**Алгоритм решения задачи.**

Создается три процесса. Первый считывает строчки и отправляет их процессу номер три. Он принимает их, обрабатывает и выводит, после чего разрешает первому процессу продолжать работу. Также первый и третий процессы передают процессу №2 данные о количестве принятых/отправленных букв. Связь между процессами осуществляется при помощи pipe`ов.

**Листинг программы**

#include <iostream>

#include <windows.h>

#include <string>

using namespace std;

string ts (int i) {

    string s = "";

    string buf = "";

    while (i > 0) {

        buf = ((i % 10) + 48);

        buf += s;

        s = buf;

        i /= 10;

    }

    return (s + "\n");

}

int main () {

    HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

    SECURITY\_ATTRIBUTES sa;

    PROCESS\_INFORMATION pi;

    STARTUPINFO si;

    sa.nLength = sizeof (SECURITY\_ATTRIBUTES);

    sa.bInheritHandle = TRUE;

    ZeroMemory (&si, sizeof(si));

    si.cb = sizeof (si);

    ZeroMemory (&pi, sizeof(pi));

    HANDLE Bin;

    HANDLE Bout;

    HANDLE Cin;

    HANDLE Cout;

    HANDLE BCin;

    HANDLE BCout;

    HANDLE sem = CreateSemaphore (

        &sa,

        0, // Start count

        1, // Max count

        NULL // Name

    );

    CreatePipe (&Bout, &Bin, &sa, 0);

    CreatePipe (&Cout, &Cin, &sa, 0);

    CreatePipe (&BCout, &BCin, &sa, 0);

    SetHandleInformation(Bin, HANDLE\_FLAG\_INHERIT, 0);

    SetHandleInformation(Cin, HANDLE\_FLAG\_INHERIT, 0);

    //SetHandleInformation(BCin, HANDLE\_FLAG\_INHERIT, 0);

    //SetHandleInformation(hConsole, HANDLE\_FLAG\_INHERIT, 0);

    si.hStdError = sem;

    si.hStdOutput = hConsole;

    si.hStdInput = Bout;

    si.dwFlags |= STARTF\_USESTDHANDLES;

    DWORD D;

    if (!CreateProcess (NULL,

        "B.exe",

        NULL,

        NULL,

        TRUE,

        0,

        NULL,

        NULL,

        &si,

        &pi)

    ) {

        printf ("ERROR\nIncorrect work of function \"CreateProcess\".");

        return -1;

    }

    si.hStdInput = Cout;

    si.hStdError = BCout;

    if (!CreateProcess (NULL,

        "C.exe",

        NULL,

        NULL,

        TRUE,

        0,

        NULL,

        NULL,

        &si,

        &pi)

    ) {

        printf ("ERROR\nIncorrect work of function \"CreateProcess\".");

        return -1;

    }

    string s = "!";

    WriteFile (Bin, &BCin, sizeof (BCin), &D, NULL);

    ReleaseSemaphore (sem, 1, NULL); // Освобождение

    while (cin >> s) {

        cout << "A: " << s << endl;

        WriteFile (Bin, s.c\_str (), s.size () + 1, &D, NULL);

        int length = s.size () + 1;

        WriteFile (Cin, &length, sizeof (length), &D, NULL);

        //int j = 6;

        //WriteFile (BCin, &j, sizeof (j), &D, NULL);

        WaitForSingleObject (sem, INFINITE); // Захват

    }

    return 0;

}

#include <iostream>

#include <windows.h>

#include <string>

using namespace std;

string ts (int i) {

    string s = "";

    string buf = "";

    while (i > 0) {

        buf = ((i % 10) + 48);

        buf += s;

        s = buf;

        i /= 10;

    }

    return (s + "\n");

}

int main () {

    HANDLE in = GetStdHandle(STD\_INPUT\_HANDLE);

    HANDLE out = GetStdHandle(STD\_OUTPUT\_HANDLE);

    HANDLE err = GetStdHandle(STD\_ERROR\_HANDLE);

    HANDLE BCin;

    DWORD D;

    long long int buffer;

    int j;

    WaitForSingleObject (err, INFINITE); // Захват

    ReadFile (in, &buffer, sizeof (long long int), &D, NULL);

    BCin = (HANDLE) buffer;

    while (1) {

        string s = "C: ";

        char c = '!';

        j = 0;

        while ((c != '\0') && (c != '\n')) {

            ReadFile (in, &c, sizeof (char), &D, NULL);

            s += c;

            j ++;

        }

        s += "\n";

        if (D == 0) {

            break;

        }

        WriteFile (out, s.c\_str (), s.size (), &D, NULL);

        WriteFile (BCin, &j, sizeof (j), &D, NULL);

        ReleaseSemaphore (err, 1, NULL); // Освобождение

    }

    return 0;

}

#include <iostream>

#include <windows.h>

#include <string>

using namespace std;

string ts (int i) {

    string s = "";

    string buf = "";

    while (i > 0) {

        buf = ((i % 10) + 48);

        buf += s;

        s = buf;

        i /= 10;

    }

    return (s + "\n");

}

int main () {

    HANDLE in = GetStdHandle(STD\_INPUT\_HANDLE);

    HANDLE out = GetStdHandle(STD\_OUTPUT\_HANDLE);

    HANDLE err = GetStdHandle(STD\_ERROR\_HANDLE);

    DWORD D;

    while (1) {

        int lA;

        int lC;

        ReadFile (in, &lA, sizeof (int), &D, NULL);

        ReadFile (err, &lC, sizeof (int), &D, NULL);

        if (D == 0) {

            break;

        }

        string buffer;

        string sA = ts (lA);

        buffer = "A: " + sA;

        sA = buffer;

        string sC = ts (lC);

        buffer = "C: " + sC;

        sC = buffer;

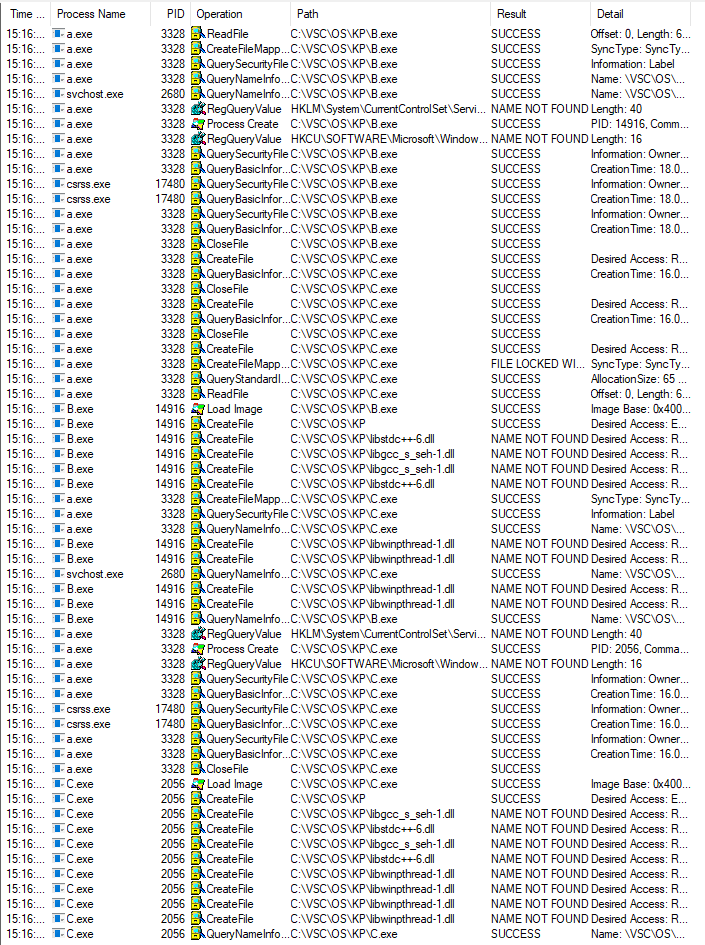
        WriteFile (out, sA.c\_str (), sizeof (sA.c\_str ()), &D, NULL);

        WriteFile (out, sC.c\_str (), sizeof (sC.c\_str ()), &D, NULL);

    }

    return 0;

}

**ProcMon**

**Выводы**

Я научился осуществлять сообщение между большим количеством процессов при помощи pipe`ов, а также использовать функции ReadFile и WriteFile для передачи Handle`ов.

**Список литературы**

1. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. —СПб.: Издательский дом «Питер», 2018. — С. 111 - 123