

Московский Авиационный Институт
(Национальный Исследовательский Университет)
Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Лабораторная работа №2 по курсу
«Операционные системы»

Тема работы
“Изучение взаимодействий между процессами”

Студент: Москвин Артём Артурович
Группа: М8О-208Б-20
Вариант: 1
Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2021

Содержание

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

Репозиторий

<https://github.com/Pert002/lab2>

Постановка задачи

Родительский процесс создает дочерний процесс. Первой строчкой пользователь в консоль родительского процесса пишет имя файла, которое будет передано при создании дочернего процесса. Родительский и дочерний процесс должны быть представлены разными программами. Родительский процесс передает команды пользователя через `pipe1`, который связан с стандартным входным потоком дочернего процесса. Дочерний процесс при необходимости передает данные в родительский процесс через `pipe2`. Результаты своей работы дочерний процесс пишет в созданный им файл. Допускается просто открыть файл и писать туда, не перенаправляя стандартный поток вывода.

Пользователь вводит команды вида: «число число число». Далее эти числа передаются от родительского процесса в дочерний. Дочерний процесс производит сложение этих чисел, а результат выводит в файл. Числа имеют тип `float`. Количество чисел может быть произвольным.

Общие сведения о программе

Вся программа содержится в одном файле `main.cpp`

Общий метод и алгоритм решения

При помощи вызова `fork` создаются два процесса.

В родительском процессе вновь вызывается `fork`, теперь активны 3 процесса – два дочерних и родительский.

Родитель считывает строки `string`, конвертирует в `c-style` строки и отправляет их по правилу сортировки в дочерние процессы. Они, в свою очередь, осуществляют требуемую операцию и выводят строки в файл.

Исходный код

Добавьте исходный код вашей программы (для вставки кода в Microsoft Word можно взять способ [здесь](#)).

main.cpp

```
#include <unistd.h>
#include <iostream>
#include <vector>
#include <string>
#include <fstream>
#include <sys/wait.h>

using namespace std;

int main() {

    int fds[2];
    if(pipe(fds) != 0) {
        perror("pipe");
    }

    int id = fork();
    if (id == -1) {
        return -1;
    }

    if (id == 0) {
        string filename;
        int len;
        read(fds[0], &len, sizeof(int));

        for (int i = 0; i < len; i++) {
            char c;
            read(fds[0], &c, sizeof(char));
            filename.push_back(c);
        }

        ofstream outfile(filename);

        int t;
        read(fds[0], &t, sizeof(int));
        int amount;
        read(fds[0], &amount, sizeof(int));

        int sum = 0;
        for (int i = 0; i < amount; i++) {
            int n;
```

```

        read(fds[0], &n, sizeof(int));
        sum += n;
    }

    for (int i = 0; i < t; i++) {
        int amount;
        read(fds[0], &amount, sizeof(int));
        int sum = 0;
        for (int i = 0; i < amount; i++) {
            int n;
            read(fds[0], &n, sizeof(int));
            sum += n;
        }
        outfile << sum << endl;
    }

    outfile.close();
    close(fds[0]);
    close(fds[1]);
}

else {
    string filename;

    cout << "Parent's PID: " << getpid() << endl;
    cout << "Child's PID: " << id << endl;
    vector<int> numbers;
    cout << "Enter the file name:\n";
    cin >> filename;

    int length = filename.length();
    write(fds[1], &length, sizeof(int));
    for (int i = 0; i < length; i++) {
        write(fds[1], &filename[i], sizeof(char));
    }

    cout << "Enter amount of commands:\n";
    int comm;
    cin >> comm;
    comm++;
    write(fds[1], &comm, sizeof(int));

    for (int i = 0; i < comm; i++)
    {
        string s;
        getline(cin, s);
        vector<int> numbers;
        string n;
    }
}

```

```

        for (int i = 0; i < s.length(); i++) {
            if ((s[i] != ' ') || (s[i] != '\0')) {
                n.push_back(s[i]);
            }
            if ((s[i] == ' ') || (s[i] == '\0') || (s[i] == '\n') || (i ==
s.length() - 1)) {
                int num = stoi(n);
                n = "";
                numbers.push_back(num);
            }
        }

        int amount = numbers.size();
        write(fds[1], &amount, sizeof(int));
        for (int i = 0; i < amount; i++) {
            write(fds[1], &numbers[i], sizeof(int));
        }
    }

    close(fds[1]);
    close(fds[0]);
}
return 0;
}

```

Демонстрация работы программы

pert@DESKTOP-L3DASJ6:/mnt/d/C C++/2 кypc/OS/lab2/src\$./a.out

Parent's PID: 37

Child's PID: 38

Enter the file name:

file

Enter amount of commands:

1

5 6 7

pert@DESKTOP-L3DASJ6:/mnt/d/C C++/2 кypc/OS/lab2/src\$ cat file

18

Выводы

Я приобрёл навыки в управлении процессами в ОС Unix и обеспечении обмена данных между процессами при помощи каналов.