

Отчёт по лабораторной работе №2
по дисциплине “Операционные системы”

Работу выполнил(а) студент(ка) группы М8О-206Б-20

Черкашин Андрей Викторович, № по списку: 27

Работа сдана: 13 ноября 2021 г.

Преподаватель: Соколов А.А.

Итоговая оценка: 5.

Подпись преподавателя _____.

1. Тема.

Процессы и потоки.

2. Цель работы.

Приобретение практических навыков в:

- управлении процессами в ОС;
- обеспечении обмена данных между процессами посредством каналов.

3. Задание.

Составить и отладить программу на языке Си, осуществляющую работу с процессами и взаимодействие между ними в одной из двух операционных систем. В результате работы программа (основной процесс) должна создать для решения задачи один или несколько дочерних процессов. Взаимодействие между процессами осуществляется через системные сигналы/события и/или каналы (pipe).

Необходимо обрабатывать системные ошибки, которые могут возникнуть в результате работы.

Группа вариантов: 5.

Родительский процесс создает два дочерних процесса. Первой строкой пользователь в консоль родительского процесса вводит имя файла, которое будет использовано для открытия File с таким именем на запись для child1.

Аналогично для второй строки и процесса child2. Родительский и дочерний процесс должны быть представлены разными программами. Родительский процесс принимает от пользователя строки произвольной длины и пересылает их в pipe1 или в pipe2 в зависимости от правила фильтрации. Процесс child1 и child2 производят работу над строками. Процессы пишут результаты своей работы в стандартный вывод.

Вариант № 22.

Правило фильтрации: с вероятностью 80% строки отправляются в pipe1, иначе в pipe2. Дочерние процессы инвертируют строки.

4. Описание программы.

Для реализации данной задачи необходимо считать со стандартного ввода названия двух файлов, затем можно записать его в массив char-ов и при открытии их на запись выдать файлам права на чтение, запись и выполнение файла. Создаем pipe и делаем 2 раза fork в родителе для создания двух дочерних процессов. В родительском пишем строки в pipe, детей же запускаем с помощью exec таким образом, чтобы для них стандартный ввод был вводом из pipe, а стандартный вывод был подменеу выводом в файл (подменяем файловые дескрипторы с помощью dup2). Распределение по pipe в родителе ведется таким образом: генерируется случайное число в пределах 10000. Если число меньше 8000, то строка отправляется в pipe1, иначе в pipe2. Дочерние программы инвертируют строки.

5. Тестирование.

Тест 1

```
a.txt
b.txt
abcd
double
int
str
```

Содержимое файла a.txt:

```
dcba
elbuod
rts
```

Содержимое файла b.txt:

```
tni
```

Тест 2

```
a.txt
```

```
b.txt
```

```
doodle
```

```
angle
```

```
rhomb
```

```
ganom
```

```
asdfsdf
```

Содержимое файла a.txt:

```
eldood
```

```
elgna
```

```
bmohr
```

```
monag
```

```
fsdfdsa
```

Содержание файла b.txt:

(пустой)

6. Листинг программы.

Находится в папке src.

7. Вывод.

В ходе выполнения данной лабораторной работы, я познакомился с созданием дочерних процессов с помощью fork. Усвоил, что при fork родительский процесс получает pid ребенка. Так же был понят принцип работы dup, что, по моему мнению, открывает больше возможностей при написании программ.