Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

Лабораторная работа №1 по курсу

«Операционные системы»

Студент: Мазепа Илья Алексеевич

Группа: М8О-209Б-23

Преподаватель: Миронов Евгений Сергеевич

Оценка: _____

Дата: _____

Подпись: _____

Москва, 2024

GitHub репозиторий: https://github.com/Tyhyqo/mai_oc

Цель работы

Приобретение практических навыков в:

– Управлении процессами в ОС

– Обеспечении обмена данных между процессами посредством каналов

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.

Вариант

Родительский процесс создает два дочерних процесса. Перенаправление стандартных потоков ввода-вывода показано на картинке выше. Child1 и Child2 можно «соединить» между собой дополнительным каналом. Родительский и дочерний процесс должны быть представлены разными программами.

Родительский процесс принимает от пользователя строки произвольной длины и пересылает их в pipe1. Процесс child1 и child2 производят работу над строками. Child2 пересылает результат своей работы родительскому процессу. Родительский процесс полученный результат выводит в стандартный поток вывода.

Child1 переводит строки в верхний регистр. Child2 превращает все пробельные символы в символ «_».

Основные технологии и моменты

Управление процессами

- Использование системного вызова fork() для создания дочерних процессов.
- Обработка системных ошибок, возникающих при создании процессов.

Взаимодействие между процессами

- Использование системного вызова pipe() для создания каналов (pipe) для обмена данными между процессами.
- Перенаправление стандартных потоков ввода-вывода с помощью системных вызовов dup2() и close().
- Использование системного вызова execve() для замены текущего процесса на новый процесс.

Обработка системных ошибок

- Проверка ошибок при создании каналов и процессов.
- Обработка ошибок при выполнении системных вызовов.

Пример работы программы

1. Родительский процесс принимает строку от пользователя.
2. Строка передается через pipe1 в первый дочерний процесс (child1).
3. Child1 переводит строку в верхний регистр и передает результат через pipe2 во второй дочерний процесс (child2).
4. Child2 заменяет все пробельные символы на символ «_» и передает результат через pipe3 обратно родительскому процессу.
5. Родительский процесс выводит результат на экран.

Пример лога работы программы

```
tyhyqo@BOOK-L939VNBBJO:~/Education/MAI/C/mai_oc/lab_1/build$ ./parent
Введите строку: sdhfsdkjfs  213131  jdakjdsj  12312  dsksflsd
Результат: SDHFSDKJFS____213131___JDAKJDSJ___12312___DSKSFLSD
```

Вывод

В результате выполнения лабораторной работы были приобретены практические навыки в управлении процессами в ОС и обеспечении обмена данных между процессами посредством каналов. Программа успешно создает два дочерних процесса,

которые обрабатывают строки, переданные родительским процессом, и возвращают результат обратно родительскому процессу для вывода.