

Межпроцессные взаимодействия (IPC) в Linux (POSIX)

Программа работы в минимальном объеме,
достаточном для оценки «удовлетворительно»

1. Управление потоками посредством сигналов

1.1. Создайте программу с порождением двух нитей так, чтобы управление второй нитью осуществлялось посредством сигнала SIGUSR1 из первой нити (после некоторого заданного интервала времени выполнения обеих нитей удалите вторую нить отправкой сигнала из первой нити).

Для этого воспользуйтесь функцией `pthread_kill(t2, SIGUSR)`; (*t2* - дескриптор второй нити). Проанализируйте полученные результаты.

Уточните заданный временной интервал и предполагаемый момент удаления нити, используя *системный таймер*.

1.2. Дополните программу, создав собственный обработчик сигнала, содержащий уведомление о начале его работы и возврат посредством функции `pthread_exit(NULL)`;

Сравните результаты, полученные после запуска этой модификации (1.2) программы с результатами предыдущей (1.1).

3. Управление линейкой сигналов (POSIX)

3.1. Используя функцию `sigaction()`, продемонстрируйте возможности управления *линейкой сигналов*, включая собственные обработчики и маскирование для разных сигналов, а также вариативность, предоставляемую структурами `sigaction (act / oldact)`.

5. Разделяемая память и средства синхронизации

5.1. Обеспечьте **разделение памяти** и необходимую **синхронизацию** между независимыми процессами так, чтобы один из процессов мог записывать информацию, а другой считывать, по мере готовности и без потери данных для обоих процессов.

5.2. Примените *различные системные средства синхронизации* для решения предыдущей задачи (п.5.1). Сравните их эффективность.

5.3. Изменится ли ситуация, если процессы заменить на *потоки одного процесса*. Приведите эксперимент.

6. Взаимодействие посредством каналов и очередей сообщений

6.1. Каналы *pipe*. Приведите примеры использования в командной строке. Приведите накладываемые ограничения на этот вид IPC, включая фрагменты системных (конфигурационных) файлов.

Реализуйте данный вид взаимодействия *программно*, используя соответствующие системные функции.

6.2. Каналы **fifo**. Реализуйте: из командной строки и *создав программу* с соответствующими системными функциями. Приведите характеристику порождаемого FIFO-файла и его *содержимого*, используя: утилиты ОС, */proc* и программно.

6.3. Осуществите передачу информации посредством очередей сообщений. Можно ли на основе этого средства организовать «почтовый ящик», т.е. не синхронизируя отправителя и получателя (без ожидания доставки).

Приведите фрагмент файла, содержащего ограничения для IPC (для каналов и сообщений).

7. Сокеты, как средство IPC ОС

Ознакомьтесь с ограничениями на реализацию сокетов, укажите, где в системе прописаны эти ограничения.

7.1. Выполните передачу информации **локально** посредством **сокетов** по **TCP/IP**

7.2. Выполните аналогичное взаимодействие на основе **UDP**. Сравните.
