

Программа лабораторной работы по курсу ОС

по теме:

Межпроцессные взаимодействия (IPC) в Linux (POSIX)

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, SIGUSR1)`; (`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).

2*. Обработчики сигналов

2. Перехватите сигнал «CTRL C» сначала однократно для процесса (2.1) и потока (2.2), а затем (2.3, 2.4) многократно с восстановлением исходного обработчика после нескольких раз срабатывания.

2.5. Приведите пример для переназначения другой комбинации клавиш.

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)`.

3.2*. Реализуйте *надежные* сигналы, организуйте эксперимент для доказательства отложенной обработки (например, в случае поступления сигнала в то время, когда уже выполняется обработка другого сигнала). Сгенерируйте ситуацию с несколькими отложенными сигналами.

4*. Сигналы реального времени

Изучаем применение 32-х сигналов второй половины полного списка сигналов (утилиты `kill -l`).

4.1. опытным путем проверьте *наличие приоритетов для сигналов* реального времени, а также 4.2. *всего списка сигналов*. Укажите сигналы с минимальным и максимальным приоритетом.

4.3. проведите эксперимент, позволяющий определить возможность организации *очереди для различных типов сигналов*, обычных (4.3.1) и реального времени (4.3.2), а также *для смешанного набора* (4.3.3);

4.4. Экспериментально подтвердите или опровергните, что обработка равно - приоритетных сигналов реального времени происходит в порядке FIFO.

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 *потoki 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 **очереди 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**

7.3*. настройте сетевое взаимодействие и повторите эксперимент в сетевом режиме (посредством сокетов TCP/IP и UDP). **Экспериментально** продемонстрируйте **разницу** между TCP и UDP реализациями.

Выполните обмен информацией для 10, 100, 1000 клиентов.

Выполнение этой работы возможно в 2-х вариантах:

минимальном (m)

полном (n - normal) для более высокой оценки по дисциплине ОС.

В минимальном объеме

пункты, **помеченные *** (звездочной), можно **НЕ выполнять**.

К ним относятся:

Раздел 2. Обработчики сигналов (п.2.1-2.5)*

П.3.2 «Надежные и ненадежные сигналы» в разделе 3*

Раздел 4 (п.4.1-4.4). Сигналы реального времени*

П.7.3. Сетевая реализация сокетов в разделе 7 Сокеты*