
Front matter

lang: ru-RU title: Именованные каналы author: Данилова Анастасия Сергеевна institute: RUDN University, Moscow, Russian Federation

date: 04.06.2021

Formatting

toc: false slide_level: 2 theme: metropolis header-includes: - \metroset{progressbar=frametitle,sectionpage=progressbar,numbering=fraction} - '\makeatletter' - '\beamer@ignorenonframefalse' - '\makeatother' aspectratio: 43

section-titles: true

Именованные каналы

Цель работы:

Приобретение практических навыков работы с именованными каналами

Межпроцессорные взаимодействия

Одним из видов взаимодействия между процессами в операционных системах является обмен сообщениями. Под сообщением понимается последовательность байтов, передаваемая от одного процесса другому.

В операционных системах типа UNIX есть 3 вида межпроцессорных взаимодействий: общепонимание (именованные каналы, сигналы), System V Interface Definition (SVID — разделяемая память, очередь сообщений, семафоры) и BSD (сокеты).

mkfifo

Для передачи данных между неродственными процессами можно использовать механизм именованных каналов (named pipes).

Именованные каналы отличаются от неименованных наличием идентификатора канала, который представлен как специальный файл (соответственно имя именованного канала — это имя файла). Поскольку файл находится на локальной файловой системе, данное IPC используется внутри одной системы.

Файлы именованных каналов создаются функцией `mkfifo`.

```
int mkfifo(const char *pathname, mode_t mode);
```

Первый параметр — имя файла, идентифицирующего канал, второй параметр — маска прав доступа к файлу. После создания файла канала процессы, участвующие в обмене данными, должны открыть этот файл либо для записи, либо для чтения. При закрытии файла сам канал продолжает существовать

Sleep()

Данная команда вводит задержку на указанное время. Команда `Sleep` имеет простой синтаксис:

```
sleep Number[Suffix]
```

Команда `sleep` без суффикса считается в секундах

Предположим, вы хотите приостановить ваш `bash`-скрипт на 5 секунд, вы можете использовать режим `sleep` следующим образом:

```
sleep 5
```

- s секунды
- m минуты
- h час
- d день

Clock

Функция `clock()` возвращает суммарное процессорное время, использованное программой. Синтаксис:

```
#include <time.h>

clock_t clock(void);
```

Пример системы клиент–сервер

Каналы представляют собой простое и удобное средство передачи данных, которое, однако, подходит не во всех ситуациях. Например, с помощью каналов довольно трудно организовать обмен асинхронными сообщениями между процессами.

Задания:

Изучите приведённые в тексте программы `server.c` и `client.c`. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения:

1. Работает не 1 клиент, а несколько (например, два).
2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию `sleep()` для приостановки работы клиента.
3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию `clock()` для определения времени работы сервера.

Выводы

Мы приобрели практические навыки работы с именованными каналами