

# ОС UNIX. Лекция 6

## Сигналы и инструменты межпроцессного взаимодействия

Михаил Пожидаев

9 октября 2023 г.

# Файловый дескриптор

## Функции для работы с файлами

Файл идентифицируется целочисленным файловым дескриптором, который может представлять не только файлы на диски, но и другие объекты, такие как сетевые соединения и трубы.

1. `open()` — открыть файл.
2. `read()/write()` — прочитать или записать бинарные данные.
3. `close()` — закрыть файл.
4. `unlink()` — удалить ссылку на файл и сам файл, если ссылка была единственной.

# Сигналы

## *Объявление обработчика сигнала*

```
volatile sig_atomic_t wasSigHup = 0;
```

```
void sigHupHandler(int r)
{
    wasSigHup = 1;
}
```

# Обработка сигнала

## *Регистрация обработчика сигнала*

```
struct sigaction sa;  
sigaction(SIGHUP, NULL, &sa);  
sa.sa_handler = sigHupHandler;  
sa.sa_flags |= SA_RESTART;  
sigaction(SIGHUP, &sa, NULL);
```

# Запретим!

## *Блокировка сигнала*

```
sigset_t blockedMask;  
sigemptyset(&blockedMask);  
sigaddset(&blockedMask, SIGHUP);  
sigprocmask(SIG_BLOCK, &blockedMask, &origMask);
```

# signalfd()

## *Пример альтернативного решения*

Функция `signalfd()` — трансляция приходящих сигналов в файловый дескриптор.

1. Сигналы для `signalfd()` могут быть также заблокированы через вызов `sigprocmask()`.
2. Полученный файловый дескриптор можно также добавлять в вызовы `select()`.

# Трубы

## Понятие трубы

Труба (pipe) — пара файловых дескрипторов, один из которых служит для записи данных, второй — для чтения.

1. Создаётся при помощи системного вызова `pipe()`.
2. Переходит в режим ожидания при отсутствии данных или при исчерпании ресурса буфера.

# Семафоры

Семафор — обобщение понятия мьютекса на случай, когда возможно вхождение в защищённый фрагмент кода более одного процесса.

Поддерживаются два варианта семафоров:

1. Семафоры System V.
2. Семафоры POSIX.



# Семафоры System V

1. Предоставляют возможность работать сразу со множеством семафоров.
2. Создаются при помощи функции `semget()`.
3. Инициализируются при помощи функции `semctl()` (отдельная инициализация является серьёзной проблемой).
4. Операции выполняются при помощи функции `semop()`.

# Семафоры POSIX

1. Являются именованными и идентифицируются символьными строками.
2. Создаются при помощи функции `sem_open()`.
3. Операции выполняются при помощи функций `sem_wait()` и `sem_post()`.

# Разделяемая память

1. Самый быстрый способ межпроцессного взаимодействия.
2. Реализуется на уровне трансляции страниц памяти в ядре операционной системы.
3. Создаётся парой функций `shmget()` и `shmat()`.

# Прочие возможности

## *Очереди сообщений и файловые блокировки*

1. Очереди сообщений (message queues) позволяют передать некоторый фрагмент данных с указанием типа сообщения из одного процесса в другой.
2. Файловые блокировки (file locks) позволяют синхронизировать работу нескольких процессов с одним и тем же открытым файлом (но не запрещают работу с ним).

# D-Bus

D-Bus — служба организации межпроцессного взаимодействия на высоком уровне.

1. Позволяет экспортировать объекты и вызывать их методы из других процессов.
2. Создаёт несколько шин, включая системную шину и сессионную для каждой пользовательской сессии.

# Спасибо за внимание!

Всё о курсе: <https://marigostra.ru/materials/unix.html>

E-mail: [msp@luwrain.org](mailto:msp@luwrain.org)

Канал в Телеграм: <https://t.me/MarigostraRu>