

Именованные каналы. Сигналы часть 1

АКОС. Лекция №14

Канал (pipe)

- Пара файловых дескрипторов:
 - на чтение
 - на запись
- Могут использовать внутри одного процесса
- Для передачи данных между разными процессами
- **Только родственные процессы**

Типы файлов в UNIX

- Регулярные
- Каталоги
- Символические ссылки
- Устройства (блочные и символьные)
- **Каналы FIFO**
- Сокеты

Что есть у файла

- Имя в файловой системе
- Атрибуты доступа

Именованные каналы

- Команда `mkfifo`
- Системный вызов `mkfifo`

Именованный канал ведет себя [почти] как обычный канал, но при этом с ним связано некоторое имя, доступное произвольному процессу

Ввод-вывод через FIFO

- Кто-то должен создать канал
- Процедура открытия является **блокирующей** (системный вызов `open`)
- Опция `O_NONBLOCK`
- Дальнейшее взаимодействие - аналогично `pipe`

Любой процесс может вмешаться в канал, если у него есть права

Демонстрации:

1) `mkfifo channel && strace cat channel`; 2) `echo hello >channel`

Ошибка broken pipe

- Попытка записи в канал, из которого никто не читает
- Ошибка EPIPE в errno после write
- Сигнал SIGPIPE

Сигнал

- Отправляется процессу другим процессом или ядром
- Возможная реакция процесса:
 - игнорировать
 - завершить свою работу
 - завершить свою работу и сделать core dump
 - приостановить работу
 - возобновить работу
 - выполнить дополнительные действия

Сигналы

Номер	Имя	По умолчанию	Описание
1	SIGHUP	Term	обрыв соединения
2	SIGINT	Term	Ctrl+C
3	SIGQUIT	Core	Ctrl+\
4	SIGILL	Core	плохая инструкция
6	SIGABRT	Core	abort()

man 7 signal

Демонстрации: 1) посылка сигналов; 2) do_abort.c

Core Dump

- Снимок памяти процесса
- Полезен для отладки
- В современных системах управляется systemd

```
/usr/sbin/sysctl kernel.core_pattern  
ulimit -c
```

Номер	Имя	По умолчанию	Описание
9	SIGKILL	Term	убийство
11	SIGSEGV	Core	что-то плохо с памятью
13	SIGPIPE	Term	Broken pipe
15	SIGTERM	Term	завершение работы
17	SIGCHLD	Ign	завершился дочерний проц.
18	SIGCONT	Cont	Команда fg
19	SIGSTOP	Stop	Ctrl+Z
23	SIGURG	Ign	Socket urgent data

Демонстрация: Ctrl+Z для tc и python

Обработка сигналов [deprecated]

```
#include <signal.h>
```

```
// Тип sighandler_t - только в Linux!  
typedef void (*sighandler_t)(int);
```

```
sighandler_t  
signal(int signum, sighandler_t handler);
```

Немного о стандартах

- Сигналы System-V (Solaris)
- Сигналы BSD (в том числе Linux)

```
gcc -std=c99    v.s.    gcc -std=gnu99
    #define _BSD_SOURCE
    #define _GNU_SOURCE
```

Обработка сигналов [deprecated]

```
#include <signal.h>
```

```
// Тип sighandler_t - только в Linux  
typedef void (*sighandler_t)(int);
```

```
// Тип sig_t - только в *BSD  
typedef void (*sig_t)(int);
```

```
sighandler_t  
signal(int signum, sighandler_t handler);
```

Сигналы BSD v.s. System-V

- В System-V обрабатывается один раз, в BSD - до отмены обработчика
- В System-V во время обработки сигнала может быть вызван обработчик другого сигнала, в BSD - блокируется до завершения обработки
- В System-V блокирующие системные вызовы завершают свою работу (EINTR в errno), в BSD - автоматически перезапускаются (кроме sleep, pause, select)

Обработка сигналов [modern-way]

```
#include <signal.h>
```

```
struct sigaction {  
    void      (*sa_handler)(int);  
    void      (*sa_sigaction)(int, siginfo_t *, void *);  
    sigset_t  sa_mask;  
    int       sa_flags;  
    void      (*sa_restorer)(void);  
};
```

```
int  
sigaction(int signum,  
          const struct sigaction *act,  
          struct sigaction *oldact);
```


Системный вызов `signal`

- Можно использовать только с обработчиками `SIG_IGN` и `SIG_DFL`
- В Linux системный вызов ведёт себя в стиле System-V
- В стандартной библиотеке Си, если объявлен макрос `_BSD_SOURCE`, то `signal` - это функция-оболочка поверх `sigaction`, а не `signal`

Обработчик сигнала

- Может быть вызван в произвольный момент времени
 - использует текущий стек (но можно в `sigaction` указать использование произвольного адреса)
 - не должен использовать высокоуровневые функции

Обработчик сигнала

- Тип данных `sig_atomic_t`
 - целочисленный (`int`)
 - выровнен по границе машинного слова
 - гарантируется атомарность чтения/записи при поступлении сигнала
- Ключевое слово `volatile`
 - указывает компилятору, что ни в коем случае нельзя оптимизировать использование переменной

Обработка SIGSEGV

- SIGSEGV - самый распространенный сигнал в случае дефектов ПО
- Его можно перехватить
- Программа уже сломана → поведение не определено
- Но можно попытаться сохранить важные данные

