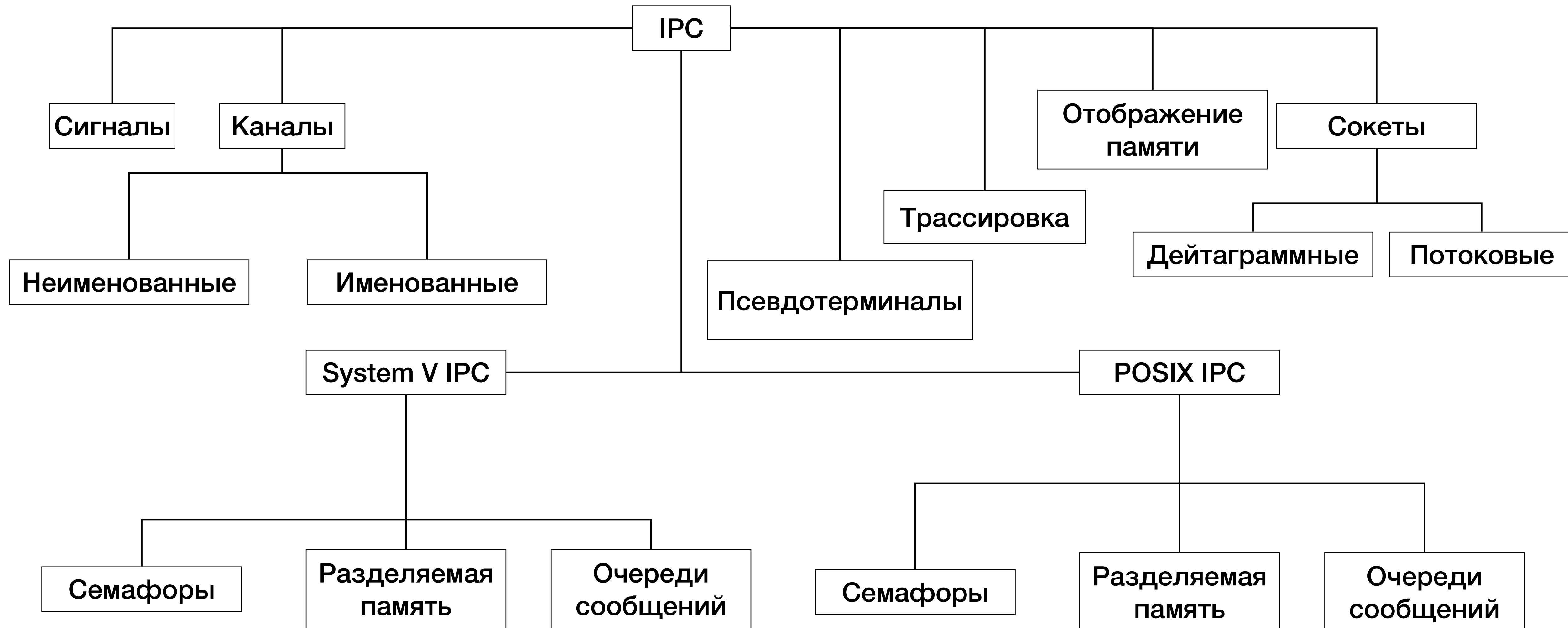


Операционные системы и сети

Взаимодействие процессов

Механизмы взаимодействия процессов

POSIX



Сигналы

- Данный механизм позволяет отправить сигнал из одного процесса в другой
- Сигнал имеет номер
- Получатель может назначить функцию обработчик сигнала
- Некоторые сигналы не могут быть перехвачены и приводят к завершению процесса (SIGKILL)

Сигналы

man signal

SIGNAL(3)

BSD Library Functions Manual

SIGNAL(3)

NAME

signal -- simplified software signal facilities

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

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

```
void (*signal(int sig, void (*func)(int)))(int);
```

or in the equivalent but easier to read typedef'd version:

```
typedef void (*sig_t) (int);
```

```
sig_t
```

```
signal(int sig, sig_t func);
```

DESCRIPTION

This signal() facility is a simplified interface to the more general sigaction(2) facility.

Сигналы

Описание сигналов

No	Name	Default Action	Description
1	SIGHUP	terminate process	terminal line hangup
2	SIGINT	terminate process	interrupt program
3	SIGQUIT	create core image	quit program
4	SIGILL	create core image	illegal instruction
5	SIGTRAP	create core image	trace trap
6	SIGABRT	create core image	abort program (formerly SIGIOT)
7	SIGEMT	create core image	emulate instruction executed
8	SIGFPE	create core image	floating-point exception
9	SIGKILL	terminate process	kill program
10	SIGBUS	create core image	bus error
11	SIGSEGV	create core image	segmentation violation
12	SIGSYS	create core image	non-existent system call invoked
13	SIGPIPE	terminate process	write on a pipe with no reader
14	SIGALRM	terminate process	real-time timer expired
15	SIGTERM	terminate process	software termination signal
16	SIGURG	discard signal	urgent condition present on socket
17	SIGSTOP	stop process	stop (cannot be caught or ignored)
18	SIGTSTP	stop process	stop signal generated from keyboard
19	SIGCONT	discard signal	continue after stop
20	SIGCHLD	discard signal	child status has changed

Сигналы

Описание сигналов (продолжение)

No	Name	Default Action	Description
21	SIGTTIN	stop process	background read attempted from control terminal
22	SIGTTOU	stop process	background write attempted to control terminal
23	SIGIO	discard signal	I/O is possible on a descriptor (see fcntl(2))
24	SIGXCPU	terminate process	cpu time limit exceeded (see setrlimit(2))
25	SIGXFSZ	terminate process	file size limit exceeded (see setrlimit(2))
26	SIGVTALRM	terminate process	virtual time alarm (see setitimer(2))
27	SIGPROF	terminate process	profiling timer alarm (see setitimer(2))
28	SIGWINCH	discard signal	Window size change
29	SIGINFO	discard signal	status request from keyboard
30	SIGUSR1	terminate process	User defined signal 1
31	SIGUSR2	terminate process	User defined signal 2

Пример. Обработчик сигнала

Отправка сигнала

man 2 kill

KILL(2)

BSD System Calls Manual

KILL(2)

NAME

kill -- send signal to a process

SYNOPSIS

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

```
int
```

```
kill(pid_t pid, int sig);
```

DESCRIPTION

The kill() function sends the signal specified by sig to pid, a process or a group of processes. Typically, Sig will be one of the signals specified in sigaction(2). A value of 0, however, will cause error checking to be performed (with no signal being sent). This can be used to check the validity of pid.

For a process to have permission to send a signal to a process designated by pid, the real or effective user ID of the receiving process must match that of the sending process or the user must have appropriate privileges (such as given by a set-user-ID program or the user is the super-user). A single exception is the signal SIGCONT, which may always be sent to any descendant of the current process.

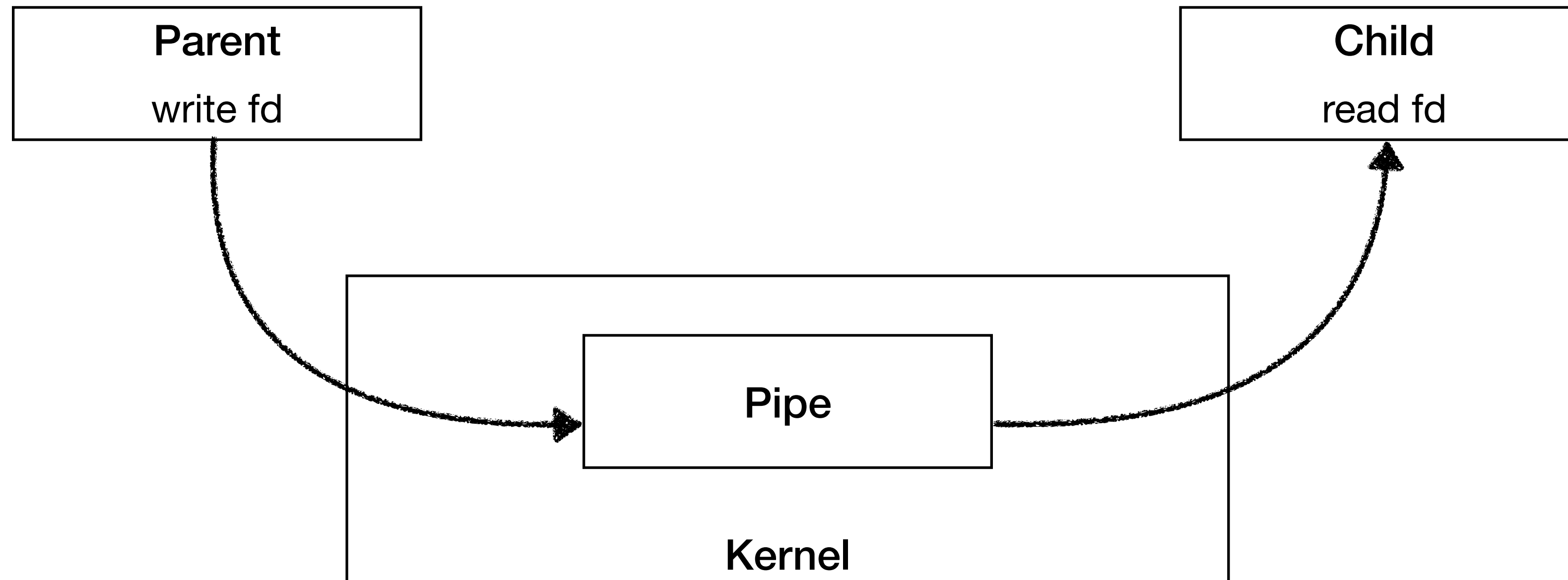
RETURN VALUES

Upon successful completion, a value of 0 is returned. Otherwise, a value of -1 is returned and errno is set to indicate the error.

Пример. Отправка сигнала

Неименованные каналы (pipe)

- Полудуплексная (однонаправленная) передачи байт
- Между родственными процессами
- Время жизни ограничено временем жизни связанных дескрипторов



Неименованные каналы

man 2 pipe

PIPE(2)

BSD System Calls Manual

PIPE(2)

NAME

pipe -- create descriptor pair for interprocess communication

SYNOPSIS

```
#include <unistd.h>
```

```
int
```

```
pipe(int fildes[2]);
```

DESCRIPTION

The pipe() function creates a pipe (an object that allows unidirectional data flow) and allocates a pair of file descriptors. The first descriptor connects to the read end of the pipe; the second connects to the write end.

Data written to fildes[1] appears on (i.e., can be read from) fildes[0]. This allows the output of one program to be sent to another program: the source's standard output is set up to be the write end of the pipe; the sink's standard input is set up to be the read end of the pipe. The pipe itself persists until all of its associated descriptors are closed.

A pipe whose read or write end has been closed is considered widowed. Writing on such a pipe causes the writing process to receive a SIGPIPE signal. Widowing a pipe is the only way to deliver end-of-file to a reader: after the reader consumes any buffered data, reading a widowed pipe returns a zero count.

Пример. Неименованный канал

ls | sort

Именованный канал FIFO

- Представлен как файл специального типа (p)
- Может быть использован любыми процессами
- Время жизни не ограничено
- Можно создать через команду `mkfifo(1)` или системный вызов `mkfifo(2)`

Именованный канал

Man 2 mkfifo

MKFIFO(2)

BSD System Calls Manual

MKFIFO(2)

NAME

mkfifo -- make a fifo file

SYNOPSIS

```
#include <sys/types.h>
```

```
#include <sys/stat.h>
```

```
int
```

```
mkfifo(const char *path, mode_t mode);
```

DESCRIPTION

mkfifo() creates a new fifo file with name path. The access permissions are specified by mode and restricted by the umask(2) of the calling process.

The fifo's owner ID is set to the process's effective user ID. The fifo's group ID is set to that of the parent directory in which it is created.

RETURN VALUES

A 0 return value indicates success. A -1 return value indicates an error, and an error code is stored in errno.

Пример. Именованный канал

System V IPC

- Семафоры - инструмент для синхронизации процессов
 - semget, semop, semctl, ftok
- Разделяемая память
 - shmget, shmat, shmdt, shmctl, ftok
- Очереди сообщений
 - msgget, msgsnd, msgrcv, msgctl

POSIX IPC

Вариация System V IPC

- POSIX Messages
 - mq_open, mq_send, mq_receive, mq_notify, mq_unlink, mq_close
- POSIX semaphores
 - sem_open, sem_init, sem_close, sem_unlink, sem_destroy, sem_post, sem_getvalue, sem_wait
- POSIX shared memory
 - shm_open, shm_unlink,

Отображение памяти

`mmap`

- Память отображаемая через `mmap` может быть доступна в нескольких процессах
- При одновременном чтении/записи необходимы синхронизировать доступ к памяти (POSIX семафоры, `msync`)

Пример*. Параллельные вычисления

Псевдотерминалы

man pty

- Псевдотерминалы (pseudoterminal - pty) - специальные псевдоустройства, которые предназначены для взаимодействия пользователя с оболочкой
- Примеры программ
 - Эмуляторы терминала: xterm, iTerm (Mac OS), Konsole, GNOME Terminal, Terminal (Mac OS)
 - Удаленный доступ: ssh, rsh, telnet
 - Мультиплексоры терминалов: screen, tmux
- Системные вызовы
 - openpty - создание нового pty
 - istty - является ли дескриптор терминалом
 - ttyname - узнать имя терминала
 - ioctl - управление различными дескрипторами, в том числе терминалами

Пример. Размер терминала

Трассировка

`man ptrace`

- Используется для отладки (debugging) и трассировки (tracing) процессов
- Системный вызов `ptrace` позволяет отправлять различные команды:
 - Включить режим трассировки
 - Выполнить один шаг
 - Продолжить выполнение
 - Подключиться/отключиться от процесса
 - Прочитать/записать значения регистров
 - Прочитать/записать значение памяти
- Точки останова (breakpoint) реализуются через перезапись машинного кода на `int 3 (0xCC)`

Примеры. Трассировка

- `gdb` - отладчик
- `strace` - трассировка системных вызовов
- `ltrace` - трассировка библиотечных вызовов

Источники

1. Обмен данными между приложениями с помощью механизма из IPC UNIX System V - ibm.com