



# Universidad Católica Argentina

Sistemas Operativos  
Lic. Ing. Osvaldo Clúa

## Signals

# Signals

- Son notificaciones que se envían a un proceso para avisar de algunos eventos.
- El comando que las envía es `kill(1)`
- El System call es `kill(2)`
- Se representan por un número y un nombre:
  - Se pueden ver con `man signal(7)`

# Signals

- En Posix 1-1990

- `SIGHUP` 1 Term Hangup detected on controlling terminal or death of controlling process
- `SIGINT` 2 Term Interrupt from keyboard
- `SIGQUIT` 3 Core Quit from keyboard
- `SIGILL` 4 Core Illegal Instruction
- `SIGABRT` 6 Core Abort signal from `abort(3)`
- `SIGFPE` 8 Core Floating point exception
- `SIGKILL` 9 Term Kill signal
- `SIGSEGV` 11 Core Invalid memory reference

# Signals

- En Posix 1-1990

- SIGPIPE 13 Term Broken pipe: write to pipe with no readers
- SIGALRM 14 Term Timer signal from alarm(2)
- SIGTERM 15 Term Termination signal
- SIGUSR1 30,10,16 Term User-defined signal 1
- SIGUSR2 31,12,17 Term User-defined signal 2
- SIGCHLD 20,17,18 Ign Child stopped or terminated

# Signals

- En Posix 1-1990

- SIGCONT 19,18,25 Cont Continue if stopped
- SIGSTOP 17,19,23 Stop Stop process
- SIGTSTP 18,20,24 Stop Stop typed at tty
- SIGTTIN 21,21,26 Stop tty input for background process
- SIGTTOU 22,22,27 Stop tty output for background process

# Signals

- En Posix 1-2001

- SIGBUS 10,7,10 Core Bus error (bad memory access)
- SIGPOLL Term Pollable event (Sys V). Synonym of SIGIO
- SIGPROF 27,27,29 Term Profiling timer expired
- SIGSYS 12,-,12 Core Bad argument to routine (SVr4)
- SIGTRAP 5 Core Trace/breakpoint trap
- SIGURG 16,23,21 Ign Urgent condition on socket (4.2BSD)
- SIGVTALRM 26,26,28 Term Virtual alarm clock (4.2BSD)
- SIGXCPU 24,24,30 Core CPU time limit exceeded (4.2BSD)
- SIGXFSZ 25,25,31 Core File size limit exceeded (4.2BSD)

# Envío de señales

- Por teclado
  - Ctl c - SIGINT
  - Ctl z SIGTSTP
  - Ctl \ SIGABRT
    - Se ven con stty -a (1)
    - Se ponen con stty

# Envío de señales

- Desde el shell
  - usando kill (1)
    - Puede enviarse el número o el nombre de la señal
- Desde un programa
  - usando kill (2)

```
#include <sys/types.h>
#include <signal.h>
int kill(pid_t pid, int sig);
```



# Atrapar señales

- Se hace con signal (2)

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

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

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

- Existe otra forma (mas moderna) usando sigaction(2).
- KILL y STOP no pueden atraparse.
- SIG\_IGN como sighandler indica ignorar la signal.
- SIG\_DFL indica volver a la acción por default.