Process management and job control

Операционни системи, ФМИ, 2019/2020

Process

- binary
- process
- thread
    - common memory
- scheduling
- context switch

Process

- PID (Process ID)
- priority & nice value
- memory
- security context
- environment
- file handles (file descriptors)

Process creation

- kernel
- `init` (PID 1)
- child process
- `ps` & `pstree`
- `fork()`
- `exec()`

Process states

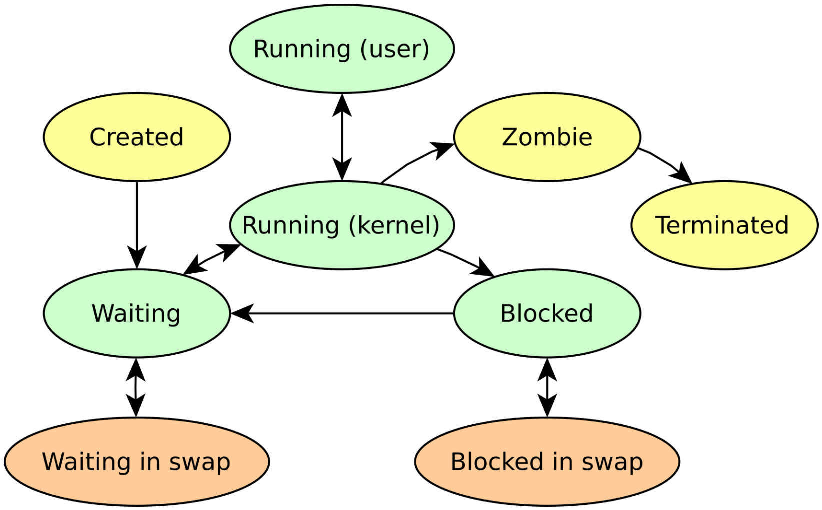

Figure 1: process_states

Process states

- **R** running/runnable (on run queue)
- **D** uninterruptable sleep (usually IO)
- **S** interruptible sleep (waiting for an event to complete)
- **T** stopped, either by a job control signal or because it is being traced
- **Z** defunct (“zombie”) process, terminated but not reaped by its parent

Process scheduling

- niceness (nice value) [-20,19]
- `nice -n 15 foo`
- `renice 15 <pid>`

Viewing processes

- `/proc`
- `ps`
  - `ps -e`
  - `ps -ef`
  - `ps -u pesho`
  - `ps -e -o user,pid`
  - `ps -u pesho -o pid=process,user=account`
  - `ps -u pesho -o pid= -o user=`
  - BSD (aux) vs. SysV (aef)
- `top`, `htop`, `atop`

Signals

- special message that can be sent to a process
- `signal(7)`
- signal vs. value
- different meanings on different architectures
- signal handlers
- some signals cannot be caught or ignored and are processed by the kernel

Signals

- SIGHUP(1)
- SIGINT(2)
- SIGQUIT(3)
- SIGKILL(9)
- SIGSEGV(11)
- SIGTERM(15)
- SIGSTOP(19)

Sending signals

- `kill <pid>`
  - SIGTERM(15) by default
  - `-l` lists all supported signals
  - `kill -KILL <pid>` or `kill -9 <pid>`
- `killall <name>`
- from keyboard
  - `Ctrl-C` - SIGINT(2)
  - `Ctrl-Z` - SIGSTOP(19)

Job control

- suspend and resume
- kernel support & user interface
- running (in foreground)
- stopped
- running in background
- SIGSTOP & SIGCONT

Job control

- `foo &`
- `Ctrl-Z` - SIGSTOP
- `jobs`
- `fg <id>`
- `bg <id>`