OPERATING SYSTEMS: SESSION6

JANUARY 2023

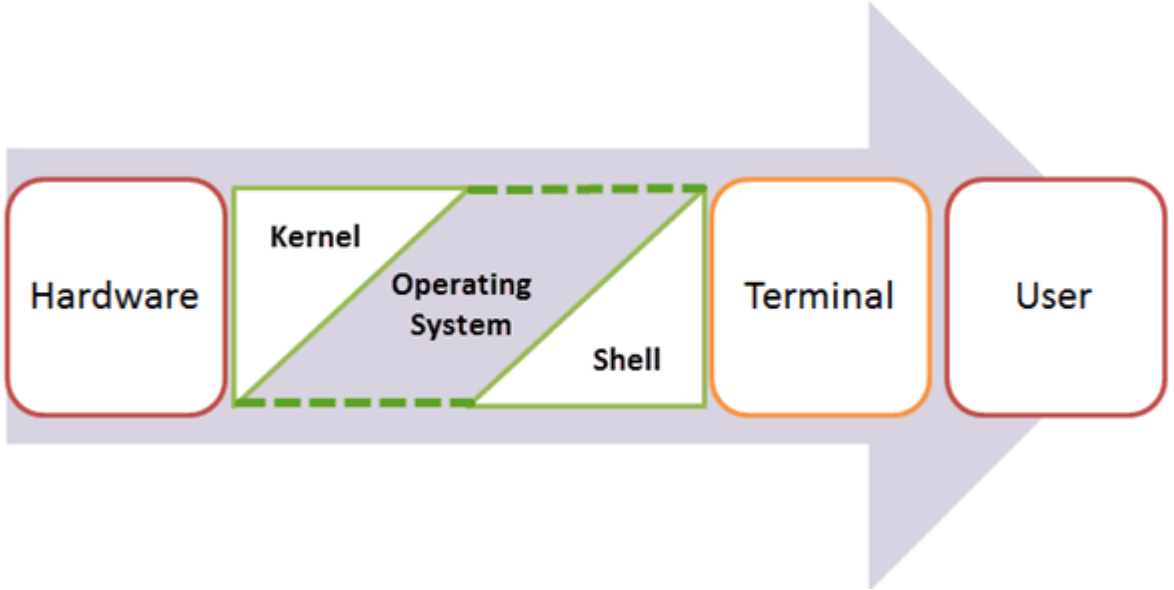
TecnoCampus Escuela Superior Politécnica

Centro adscrito a la upf. Universitat Pompeu Fabra Barcelona

TecnoCampus 10 años

OS FEATURES: KERNEL?

- The kernel is the **central** component of a computer operating system.
- The only job performed by the kernel is to **manage the communication between the software and the hardware.** The Kernel is at the nucleus of a computer. It makes the communication between the hardware and software possible.
- While the **Kernel** is the innermost part of an operating system, a **shell** is the outermost one.



OS FEATURES: FEATURES OF KERNEL

- LOW-LEVEL SCHEDULING OF PROCESSES
- INTER-PROCESS COMMUNICATION
- PROCESS SYNCHRONIZATION
- CONTEXT SWITCHING

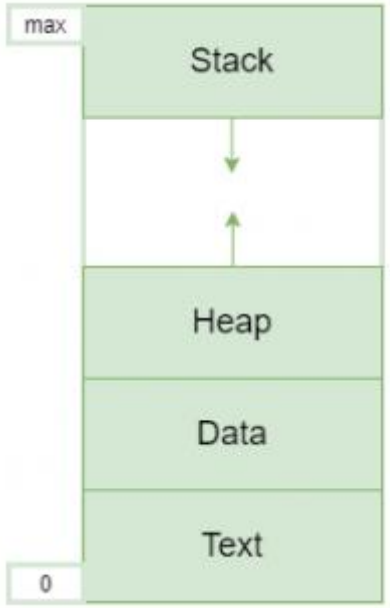
OS FEATURES: PROGRAM/PROCESS EXECUTION

- Which is the way a process can be executed on the system?
- Why do we need to execute a process?

- Remember what is a PROCESS?
  - A program in execution on the system

HEAP: Allocates memory (dynamic memory)

DATA: Variables (global variables)



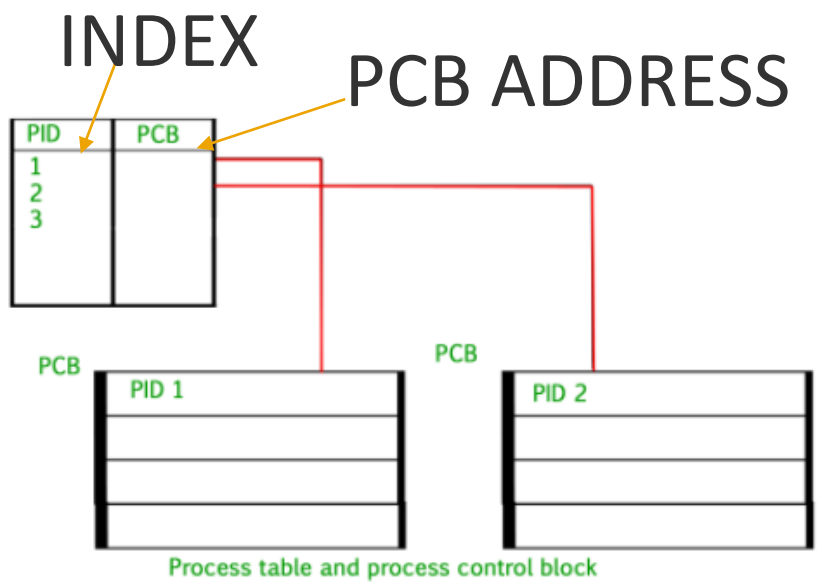
STACK: stores temporary data (parameters, return addresses, local variables)

TEXT: Code

- What does the OS need to run a “process”?

PROCESS MANAGEMENT→PROCESS CONTROL BLOCK

- The information the OS needs to run a process is stored on **PROCESS CONTROL BLOCK (PCB):** a kind of repository of information associated with the process
- Can also be known as:
  - TASK CONTROL BLOCK
  - ENTRY IN THE PROCESS TABLE
  - …
- The PCB is written in protected memory
- The OS has a table with a pointer per PCB

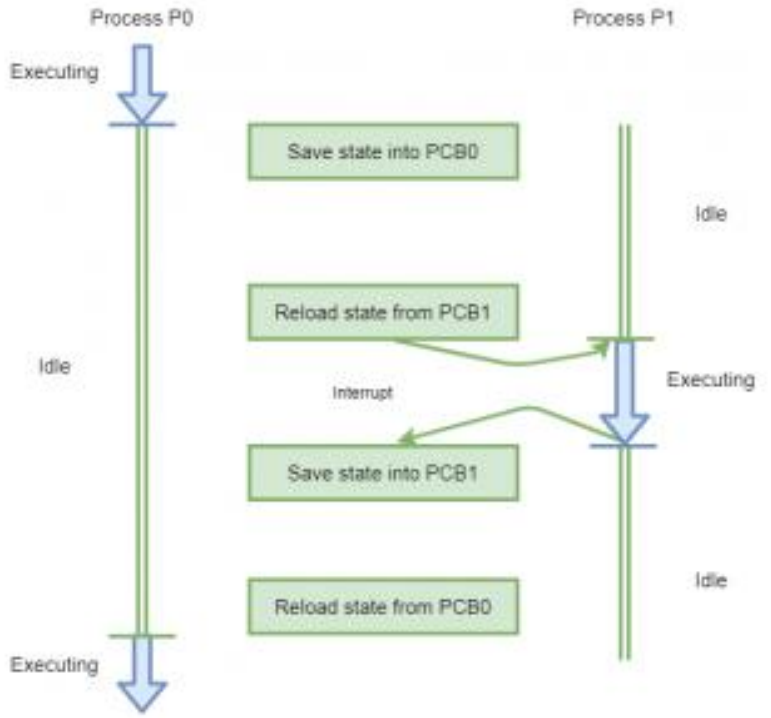


PROCESS MANAGEMENT: CONTEXT SWITCHING

- When the OS must stop a process and start running a “new” process
  - STORE ALL RELEVANT INFORMATION FROM THE PROCESS UNDER EXECUTION
  - LOAD ALL RELEVANT INFORMATION FROM THE NEW PROCESS FROM THE LAST EXECUTION OF THE PROCESS

PROCESS MANAGEMENT→CONTEXT SWITCHING

- What must be done if OS wants to switch from P0 to P1:
  - STEP 1: stop P0
  - STEP 2: save context P0→PCB0
  - STEP 3: restore context P1→PCB1
- The OS is now ready to execute P1



PROCESS MANAGEMENT→PROCESS CONTROL BLOCK

- STRUCTURE OF THE PCB:
  - PROCESS STATE: ¿?
  - PROCESS NUMBER: one process one PID
  - PROGRAM COUNTER (PC):

  Where the process is executing the program
  - REGISTERS:

  Which is the value of the MP's registers
  - LIST OF OPEN FILES:

  Which are the FILE-DESCRIPTORS of files…
  - ACCOUNTING AND EXTRA INFORMATION…

Process Control Block (PCB)

PROCESS MANAGEMENT→PROCESS CONTROL BLOCK

- ACCOUNTING AND EXTRA INFORMATION…
  - CPU SCHEDULING INFORMATION:
    - Process priority
    - Pointer to Scheduling Queues
  - MEMORY MANAGEMENT INFORMATION:
    - Page Table (PAGINATION)
    - Segment Table (SEGMENTATION)
  - ACCOUNTING INFORMATION:
    - Amount of CPU used
    - Time limits
    - Accounting figures
    - Job/Process/Threads figures
  - I/O STATUS INFORMATION
  - SYNCRHONIZATION INFORMATION

```
devasc@LJG:~$ time ls
Desktop  Documents  Downloads  labs  pt  snap

real    0m0.012s
user    0m0.005s
sys     0m0.006s
```

- Processes information: **/proc** folder

```
SO: pwd
/proc
SO: ls
1    222   buddyinfo   crypto       fs           key-users    locks      net            stat          uptime
110  231   bus         devices      interrupts   keys         mdstat     pagetypeinfo   swaps         version
111  438   cgroups     diskstats    iomem        kmsg         meminfo    partitions     sys           vmallocinfo
112  439   cmdline     dma          ioports      kpagecgroup  misc       sched_debug    sysvipc       vmstat
121  440   config.gz   driver       irq          kpagecount   modules    schedstat      thread-self   zoneinfo
154  538   consoles    execdomains  kallsyms     kpageflags   mounts     self           timer_list
221  acpi  cpuinfo     filesystems  kcore        loadavg      mtrr       softirqs       tty
SO:
```

- Processes information: **/proc/stat** file

```
SO: more stat
cpu  60038 0 87257 137701594 66691 0 1622 0 0 0
cpu0 14951 0 19214 34401464 41300 0 994 0 0 0
cpu1 14998 0 25337 34430760 7544 0 474 0 0 0
cpu2 15044 0 16924 34440492 7136 0 72 0 0 0
cpu3 15045 0 25782 34428876 10709 0 82 0 0 0
intr 4360527 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ctxt 22697728
btime 1641463108
processes 2849
procs_running 1
procs_blocked 0
softirq 10213181 0 2931870 12 226659 3054 0 148298 4417455 0 2485833
SO: _
```

AGGREGATED INFORMATION FOR ALL THE CPUs REPORTED ON **/proc/cpuinfo**
The numbers behind the "cpu" lines identify the amount of time the CPU has spent performing different kind of work. The time is measured in USER_HZ (also called **JIFFIES**) 1/100th of a second

```
SO: getconf CLK_TCK
100
SO: _
```

The first column of the "**intr**" line is the total of all interrupts served on the system since boot time. The following counters are the count for each possible system interrupt.

PROCESS MANAGEMENT→PROCESS CONTROL BLOCK

- Processes information: **PID** folder
  - PID=440→the **bash** session we are executing

```
SO: ps
  PID TTY          TIME CMD
  440 pts/1    00:00:00 bash
  558 pts/1    00:00:00 ps
SO: pwd
/proc/440
SO: ls
arch_status   comm             fd          maps        ns             projid_map   smaps_rollup   task
attr          coredump_filter  fdinfo      mem         oom_adj        root         stack          timers
auxv          cpuset           gid_map     mountinfo   oom_score      sched        stat           timerslack_ns
cgroup        cwd              io          mounts      oom_score_adj  schedstat    statm          uid_map
clear_refs    environ          limits      mountstats  pagemap        setgroups    status         wchan
cmdline       exe              map_files   net         personality    smaps        syscall
SO:
```

PROCESS MANAGEMENT→PROCESS CONTROL BLOCK

- Processes information: **PID** folder
  - PID=440→the **bash** session we are executing: **environ**

```
SO: more 440/environ
HOSTTYPE=x86_64mes:/usr/lib/wsl/lib:/mnt/c/Program Files/WindowsApps/CanonicalGroupLimited.Ubuntu20.04onWindows_2004.
_79rhkp1fndgsc:/mnt/c/Program Files (x86)/VMware/VMware Player/bin/:/mnt/c/WINDOWS/system32:/mnt/c/WINDOWS:/mnt/c/WIN
DOWS/System32/Wbem:/mnt/c/WINDOWS/System32/WindowsPowerShell/v1.0/:/mnt/c/WINDOWS/System32/OpenSSH/:/mnt/c/Program Fi
les/Docker/Docker/resources/bin:/mnt/c/ProgramData/DockerDesktop/version-bin:/mnt/c/Program Files (x86)/GnuPG/bin:/mn
t/c/Users/leonard/AppData/Local/Microsoft/WindowsApps:/mnt/c/Program Files/Multipass/bin_INTEROP=/run/WSL/439_interop
SO: _
```

  - PID=440→the **bash** session we are executing: **cmdline**

```
SO: more cmdline
-bash
SO: _
```

- Processes information: **PID** folder
  - PID=440→the **bash** session we are executing: **limits**

```
SO: more 440/limits
Limit                     Soft Limit           Hard Limit           Units
Max cpu time              unlimited            unlimited            seconds
Max file size             unlimited            unlimited            bytes
Max data size             unlimited            unlimited            bytes
Max stack size            8388608              unlimited            bytes
Max core file size        0                    unlimited            bytes
Max resident set          unlimited            unlimited            bytes
Max processes             50700                50700                processes
Max open files            1024                 4096                 files
Max locked memory         65536                65536                bytes
Max address space         unlimited            unlimited            bytes
Max file locks            unlimited            unlimited            locks
Max pending signals       50700                50700                signals
Max msgqueue size         819200               819200               bytes
Max nice priority         0                    0
Max realtime priority     0                    0
Max realtime timeout      unlimited            unlimited            us
SO:
```

PROCESS MANAGEMENT→PROCESS CONTROL BLOCK

- **sched**

Time spent on the CPU

Context Switching

Process Priority on System

Time spent on the CPU

```
SO: more sched
bash (440, #threads: 1)
-------------------------------------------------------------------
se.exec_start                                :      341222521.840600
se.vruntime                                  :        3143319.319747
se.sum_exec_runtime                          :           401.973400
se.nr_migrations                             :                   11
nr_switches                                  :                  840
nr_voluntary_switches                        :                  830
nr_involuntary_switches                      :                   10
se.load.weight                               :              1048576
se.avg.load_sum                              :                 1600
se.avg.runnable_sum                          :              1638400
se.avg.util_sum                              :              1638400
se.avg.load_avg                              :                   34
se.avg.runnable_avg                          :                   34
se.avg.util_avg                              :                   34
se.avg.last_update_time                      :       341222521839616
se.avg.util_est.ewma                         :                   34
se.avg.util_est.enqueued                     :                   34
policy                                       :                    0
prio                                         :                  120
clock-delta                                  :                  200
SO: more schedstat
404110600 933700 844
SO:
```

Time spent waiting on a run queue

# of time slices run on this CPU

PROCESS MANAGEMENT→PROCESS CONTROL BLOCK

```
SO: for f in `ls -d [0-9]*/`; do echo $f; more $f"sched"|grep prio; done
1/
prio                                 :               120
1009/
more: stat of 1009/sched failed: No such file or directory
110/
prio                                 :               120
111/
prio                                 :               120
112/
prio                                 :               120
121/
prio                                 :               120
154/
prio                                 :               120
221/
prio                                 :               120
222/
prio                                 :               120
231/
prio                                 :               120
438/
prio                                 :               120
439/
prio                                 :               120
440/
prio                                 :               120
SO:
```

PRIORITY:
0-99→REAL TIME
100-139→CFS SCHEDULING

- There are a total of **140** priorities and **TWO** distinct priority ranges implemented in Linux
  - **NICE VALUE** (NICENESS): ranges from **-20 (HIGHEST PRIORITY VALUE)** to **19 (LOWEST PRIORITY VALUE)**. Default=0
  - **REAL-TIME PRIORITY**: ranges from **0** to **139**

```
SO: ps -eo pid,ppid,ni,comm
  PID  PPID  NI COMMAND
    1     0   0 init
  110     1   0 init
  111   110   0 init
  112   111   0 docker-desktop-
  121   110   0 init <defunct>
  154     1   0 init
  221   154   0 sudo
  222   221   0 dockerd
  231   222   0 containerd
  438     1   0 init
  439   438   0 init
  440   439   0 bash
 1159   440   0 ps
SO:
```

- YOU CAN MOVE FROM NICE→PRIO

```
Total number of priorities = 140
Real time priority range(PR or PRI):  0 to 99
User space priority range: 100 to 139
```

- NICE VALUE RANGE (NI): -20 TO 19

```
PR = 20 + NI
PR = 20 + (-20 to + 19)
PR = 20 + -20  to 20 + 19
PR = 0 to 39 which is same as 100 to 139.
```

- PRIO=120→PR=20→NICE=0

PROCESS MANAGEMENT→PROCESS CONTROL BLOCK

SO: top

```
top - 10:32:01 up 3 days, 23:33,  0 users,  load average: 0.00, 0.00, 0.00
Tasks:  13 total,   1 running,  11 sleeping,   0 stopped,   1 zombie
%Cpu(s):  0.0 us,  0.1 sy,  0.0 ni, 99.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  12679.2 total,  10597.2 free,    693.8 used,   1388.2 buff/cache
MiB Swap:   4096.0 total,   4096.0 free,      0.0 used.  11377.5 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1 root      20   0     900    528    464 S   0.0   0.0   0:00.04 init
  110 root      20   0     900     84     20 S   0.0   0.0   0:00.00 init
  111 root      20   0     900     84     20 S   0.0   0.0   0:00.01 init
  112 root      20   0 1309876  28140  12304 S   0.0   0.2   0:11.46 docker-desktop-
  121 root      20   0       0      0      0 Z   0.0   0.0   0:00.00 init
  154 root      20   0     900     84     20 S   0.0   0.0   0:00.03 init
  221 root      20   0   11024   4568   3880 S   0.0   0.0   0:00.01 sudo
  222 root      20   0 1168608  96664  47660 S   0.0   0.7   0:38.14 dockerd
  231 root      20   0 1198376  59468  27660 S   0.0   0.5   0:41.92 containerd
  438 root      20   0     900     84     20 S   0.0   0.0   0:00.00 init
  439 root      20   0     900     84     20 S   0.0   0.0   0:00.46 init
  440 ljg       20   0   10300   5428   3532 S   0.0   0.0   0:01.13 bash
 1160 ljg       20   0   10876   3700   3188 R   0.0   0.0   0:00.01 top
```

Time spent on the CPU

PROCESS MANAGEMENT→PROCESS CONTROL BLOCK

SO: htop

```
  1  [                                          0.0%]   Tasks: 12, 36 thr; 1 running
  2  [                                          0.0%]   Load average: 0.03 0.02 0.00
  3  [                                          0.0%]   Uptime: 3 days, 23:35:50
  4  [                                          0.0%]
Mem[||||||||||                          1.04G/12.4G]
Swp[                                       0K/4.00G]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 1161 ljg        20   0  8148  3676  3028 R  0.0  0.0  0:00.13 htop
    6 root       20   0   900   528   464 S  0.0  0.0  0:00.00 /init
    1 root       20   0   900   528   464 S  0.0  0.0  0:00.04 /init
  110 root       20   0   900    84    20 S  0.0  0.0  0:00.00 /init
  111 root       20   0   900    84    20 S  0.0  0.0  0:00.01 /init
  113 root       20   0 1279M 28140 12304 S  0.0  0.2  0:00.72 /mnt/wsl/docker-desktop/docker-desktop-proxy --distro-
  114 root       20   0 1279M 28140 12304 S  0.0  0.2  0:00.23 /mnt/wsl/docker-desktop/docker-desktop-proxy --distro-
  115 root       20   0 1279M 28140 12304 S  0.0  0.2  0:02.48 /mnt/wsl/docker-desktop/docker-desktop-proxy --distro-
  116 root       20   0 1279M 28140 12304 S  0.0  0.2  0:01.66 /mnt/wsl/docker-desktop/docker-desktop-proxy --distro-
  117 root       20   0 1279M 28140 12304 S  0.0  0.2  0:00.00 /mnt/wsl/docker-desktop/docker-desktop-proxy --distro-
  118 root       20   0 1279M 28140 12304 S  0.0  0.2  0:02.20 /mnt/wsl/docker-desktop/docker-desktop-proxy --distro-
  119 root       20   0 1279M 28140 12304 S  0.0  0.2  0:01.61 /mnt/wsl/docker-desktop/docker-desktop-proxy --distro-
  120 root       20   0 1279M 28140 12304 S  0.0  0.2  0:02.47 /mnt/wsl/docker-desktop/docker-desktop-proxy --distro-
  112 root       20   0 1279M 28140 12304 S  0.0  0.2  0:11.46 /mnt/wsl/docker-desktop/docker-desktop-proxy --distro-
  154 root       20   0   900    84    20 S  0.0  0.0  0:00.03 /init
  221 root       20   0 11024  4568  3880 S  0.0  0.0  0:00.01 sudo dockerd
  223 root       20   0 1141M 96664 47660 S  0.0  0.7  0:02.37 dockerd
  224 root       20   0 1141M 96664 47660 S  0.0  0.7  0:03.50 dockerd
  225 root       20   0 1141M 96664 47660 S  0.0  0.7  0:00.00 dockerd
  226 root       20   0 1141M 96664 47660 S  0.0  0.7  0:02.85 dockerd
F1Help F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Time spent on the CPU

The process state represents a condition of the process at a specific instant of time

- **NEW**
- **READY**
- **RUNNING**
- **WAITING**
- **TERMINATED**

- **NEW**: when the process is **created** (a program moves to execution for the first time)→PCB is created, and process moved to **READY** (because the process has been admitted by the OS)(maximum number of processes to be executed on the system)

- **PID_MAX**: **/proc/sys/kernel/pid_max** has nothing to do with the maximum number of processes that can be run at any given time. It is, in fact, **the maximum numerical PROCESS IDENTIFIER than can be assigned by the kernel**.
- **THREADS_MAX**: **/proc/sys/kernel/threads-max** is actually the maximum number of elements contained in the data structure **task_struct**. Which is the data structure that contains the list of processes, or as they can be called, **tasks**, so the max number of threads/processes on the system
- **ULIMIT**: **ulimit** is, as the name implies, a **per-user limit**. The -u flag is defined as "The maximum number of processes available to a single user"

PROCESS MANAGEMENT→PROCESS STATE

```
devasc@LJG:~$ sysctl -a | grep kernel.pid_max
sysctl: permission denied on key 'fs.protected_fifos'
sysctl: permission denied on key 'fs.protected_hardlinks'
sysctl: permission denied on key 'fs.protected_regular'
sysctl: permission denied on key 'fs.protected_symlinks'
sysctl: permission denied on key 'kernel.cad_pid'
kernel.pid_max = 4194304
sysctl: permission denied on key 'kernel.unprivileged_userns_apparmor_policy'
sysctl: permission denied on key 'kernel.usermodehelper.bset'
sysctl: permission denied on key 'kernel.usermodehelper.inheritable'
sysctl: permission denied on key 'net.core.bpf_jit_harden'
sysctl: permission denied on key 'net.core.bpf_jit_kallsyms'
sysctl: permission denied on key 'net.core.bpf_jit_limit'
sysctl: permission denied on key 'net.ipv4.tcp_fastopen_key'
sysctl: permission denied on key 'net.ipv6.conf.all.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.default.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.dummy0.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.enp0s3.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.enp0s8.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.lo.stable_secret'
sysctl: permission denied on key 'vm.mmap_rnd_bits'
sysctl: permission denied on key 'vm.mmap_rnd_compat_bits'
sysctl: permission denied on key 'vm.stat_refresh'
devasc@LJG:~$
```

64-bit systems (up to 2^{22} = 4,194,304)

```
ljg@TEL-Leonard:~$ cat /proc/sys/kernel/pid_max
32768
ljg@TEL-Leonard:~$
```

```
devasc@LJG:~$ cat /proc/sys/kernel/pid_max
4194304
devasc@LJG:~$
```

```
devasc@LJG:~$ sudo sysctl kernel.pid_max=4000000
kernel.pid_max = 4000000
devasc@LJG:~$ sysctl -a | grep kernel.pid_max
sysctl: permission denied on key 'fs.protected_fifos'
sysctl: permission denied on key 'fs.protected_hardlinks'
sysctl: permission denied on key 'fs.protected_regular'
sysctl: permission denied on key 'fs.protected_symlinks'
sysctl: permission denied on key 'kernel.cad_pid'
kernel.pid_max = 4000000
```

PROCESS MANAGEMENT→PROCESS STATE

```
devasc@LJG:~$ sysctl -a | grep kernel.threads-max
sysctl: permission denied on key 'fs.protected_fifos'
sysctl: permission denied on key 'fs.protected_hardlinks'
sysctl: permission denied on key 'fs.protected_regular'
sysctl: permission denied on key 'fs.protected_symlinks'
sysctl: permission denied on key 'kernel.cad_pid'
kernel.threads-max = 30851
sysctl: permission denied on key 'kernel.unprivileged_userns_apparmor_policy'
sysctl: permission denied on key 'kernel.usermodehelper.bset'
sysctl: permission denied on key 'kernel.usermodehelper.inheritable'
sysctl: permission denied on key 'net.core.bpf_jit_harden'
sysctl: permission denied on key 'net.core.bpf_jit_kallsyms'
sysctl: permission denied on key 'net.core.bpf_jit_limit'
sysctl: permission denied on key 'net.ipv4.tcp_fastopen_key'
sysctl: permission denied on key 'net.ipv6.conf.all.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.default.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.dummy0.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.enp0s3.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.enp0s8.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.lo.stable_secret'
sysctl: permission denied on key 'vm.mmap_rnd_bits'
sysctl: permission denied on key 'vm.mmap_rnd_compat_bits'
sysctl: permission denied on key 'vm.stat_refresh'
devasc@LJG:~$
```

- Limitations per user/group…

```
devasc@LJG:~$ ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 15425
max locked memory       (kbytes, -l) 65536
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes              (-u) 15425
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
devasc@LJG:~$
```

$30851 \div 2 = 15425$

PROCESS MANAGEMENT→PROCESS STATE

- Limitations per user/group…
- /etc/security/limits.conf

etc > security > limits.conf

```
# /etc/security/limits.conf
#
#Each line describes a limit for a user in the form:
#
#<domain>        <type>  <item>  <value>
#
#Where:
#<domain> can be:
#        - a user name
#        - a group name, with @group syntax
#        - the wildcard *, for default entry
#        - the wildcard %, can be also used with %group syntax,
#                 for maxlogin limit
#        - NOTE: group and wildcard limits are not applied to root.
#          To apply a limit to the root user, <domain> must be
#          the literal username root.
#
#<type> can have the two values:
#        - "soft" for enforcing the soft limits
#        - "hard" for enforcing hard limits
#
#<item> can be one of the following:
#        - core - limits the core file size (KB)
#        - data - max data size (KB)
#        - fsize - maximum filesize (KB)
#        - memlock - max locked-in-memory address space (KB)
#        - nofile - max number of open file descriptors
#        - rss - max resident set size (KB)
#        - stack - max stack size (KB)
#        - cpu - max CPU time (MIN)
#        - nproc - max number of processes
#        - as - address space limit (KB)
#        - maxlogins - max number of logins for this user
#        - maxsyslogins - max number of logins on the system
#        - priority - the priority to run user process with
#        - locks - max number of file locks the user can hold
#        - sigpending - max number of pending signals
#        - msgqueue - max memory used by POSIX message queues (bytes)
#        - nice - max nice priority allowed to raise to values: [-20, 19]
#        - rtprio - max realtime priority
#        - chroot - change root to directory (Debian-specific)
#
#<domain>      <type>  <item>         <value>
#

#*               soft    core            0
#root            hard    core            100000
#*               hard    rss             10000
#@student        hard    nproc           20
#@faculty        soft    nproc           20
#@faculty        hard    nproc           50
#ftp             hard    nproc           0
#ftp             -       chroot          /ftp
#@student        -       maxlogins       4

# End of file

```

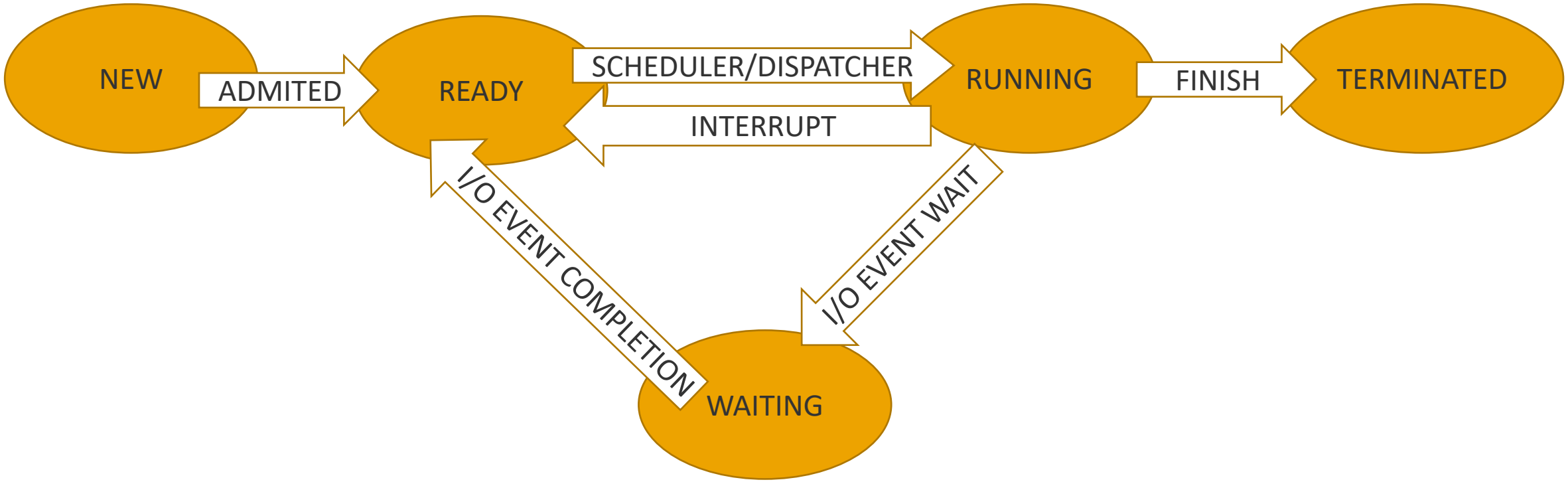
- **READY**:
  - The process could be executed
  - The process is waiting the OS to take the decision that process would be the next process to be executed
  - The process is then moved to primary memory…. So the execution will be better performed
  - **THE PROCESS IS NOT ALREADY EXECUTED**.
  - **THE PROCESS IS IN QUEUE TO BE MOVED TO RUNNING STATE**

- **RUNNING**
  - The process is under execution
  - Depending on the number of core/system you can have more than one process in that state

- **WAITING**: the process is moved from RUNNING to that state. The process can not be executed due to some reason
  - The process is waiting for the **allocation** of some resources
  - The process is waiting for an **event** (like I/O operations) to be completed
  - The process is waiting due to a **synchronization** mechanism
- Before moving to RUNNING the process must be moved to READY

- **TERMINATED**: the process is finished, and will deallocate all the resources, and PCB will be removed from table, and process will never be executed anymore

PROCESS MANAGEMENT→PROCESS STATE



PROCESS MANAGEMENT→PROCESS STATE

NEW	READY	RUNNING	WAITING	TERMINATED
1				
	1			
		1		
				1

PROCESS MANAGEMENT→PROCESS STATE

NEW	READY	RUNNING	WAITING	TERMINATED
1,2, 3				
	1, 2, 3			
	2, 3	1		
	1, 3	2		
	1	3	2	
	1, 2	3		

PROCESS MANAGEMENT→DISPATCHER/SCHEDULER

- **SCHEDULER**: choose/select based on scheduling algorithms which will be the next process (next process to be moved to RUNNING STATE from READY STATE)
- **DISPATCHER**: perform the context switching

PROCESS MANAGEMENT→PROCESS ID

- A kind of INDEX, IDENTIFIER
- Can be used by the system/other processes to communicate with the process

```
devasc@LJG:~$ ps --help

Usage:
 ps [options]

 Try 'ps --help <simple|list|output|threads|misc|all>'
  or 'ps --help <s|l|o|t|m|a>'
 for additional help text.

For more details see ps(1).
devasc@LJG:~$
```

```
File  Edit  View  Search  Terminal  Help
PS(1)                          User Commands                          PS(1)

NAME
       ps - report a snapshot of the current processes.

SYNOPSIS
       ps [options]

DESCRIPTION
       ps displays information about a selection of the active processes.  If
       you want a repetitive update of the selection and the displayed
       information, use top(1) instead.

       This version of ps accepts several kinds of options:

       1   UNIX options, which may be grouped and must be preceded by a dash.
       2   BSD options, which may be grouped and must not be used with a dash.
       3   GNU long options, which are preceded by two dashes.

       Options of different types may be freely mixed, but conflicts can
       appear.  There are some synonymous options, which are functionally
       identical, due to the many standards and ps implementations that this
       ps is compatible with.
Manual page ps(1) line 1 (press h for help or q to quit)
```

PROCESS MANAGEMENT→PROCESS ID

```
devasc@LJG: ~
File  Edit  View  Search  Terminal  Help
   2976 pts/1    00:00:00 ps
devasc@LJG:~$ ps -ejH
    PID    PGID     SID TTY          TIME CMD
      2       0       0 ?        00:00:00 kthreadd
      3       0       0 ?        00:00:00   rcu_gp
      4       0       0 ?        00:00:00   rcu_par_gp
      6       0       0 ?        00:00:00   kworker/0:0H-kblockd
      9       0       0 ?        00:00:00   mm_percpu_wq
     10       0       0 ?        00:00:00   ksoftirqd/0
     11       0       0 ?        00:00:06   rcu_sched
     12       0       0 ?        00:00:00   migration/0
     13       0       0 ?        00:00:00   idle_inject/0
     14       0       0 ?        00:00:00   cpuhp/0
     15       0       0 ?        00:00:00   cpuhp/1
     16       0       0 ?        00:00:00   idle_inject/1
     17       0       0 ?        00:00:02   migration/1
     18       0       0 ?        00:00:00   ksoftirqd/1
     20       0       0 ?        00:00:00   kworker/1:0H-kblockd
     21       0       0 ?        00:00:00   kdevtmpfs
     22       0       0 ?        00:00:00   netns
     23       0       0 ?        00:00:00   rcu_tasks_kthre
     24       0       0 ?        00:00:00   kauditd
     25       0       0 ?        00:00:00   khungtaskd
     26       0       0 ?        00:00:00   oom_reaper
```

```
devasc@LJG: ~
File  Edit  View  Search  Terminal  Help
       other users or not on a terminal.  These effects are not considered
       when options are described as being "identical" below, so -M will be
       considered identical to Z and so on.

       Except as described below, process selection options are additive.  The
       default selection is discarded, and then the selected processes are
       added to the set of processes to be displayed.  A process will thus be
       shown if it meets any of the given selection criteria.

EXAMPLES
       To see every process on the system using standard syntax:
          ps -e
          ps -ef
          ps -eF
          ps -ely

       To see every process on the system using BSD syntax:
          ps ax
          ps axu

       To print a process tree:
          ps -ejH
          ps axjf
Manual page ps(1) line 47 (press h for help or q to quit)
```

```
  2218    1540    1540 ?        00:00:17   mate-terminal
  2225    2225    2225 pts/0    00:00:00     bash
  2929    2929    2225 pts/0    00:00:00       man
  2939    2929    2225 pts/0    00:00:00         pager
  2970    2970    2970 pts/1    00:00:00     bash
  2977    2977    2970 pts/1    00:00:00       ps
```

```
ljg@TEL-Leonard:~$ ps -ejH
  PID  PGID   SID TTY          TIME CMD
    1     1     1 ?        00:00:00 init
    7     7     7 tty1     00:00:00   init
    8     8     7 tty1     00:00:00     bash
   85    85     7 tty1     00:00:00       ps
ljg@TEL-Leonard:~$ pidof bash
8
ljg@TEL-Leonard:~$
```

```
ljg@TEL-Leonard:~$ echo $$
8
ljg@TEL-Leonard:~$ echo $PPID
7
ljg@TEL-Leonard:~$
```

Environment variable

- PID = 0 → **SWAPPER/SCHEDULER**
- PID = 1 → **INIT**: responsible for starting and shutting down the system
  It is started by the kernel itself (it has not a parent process)
  It functions as an adoptive parent for all orphaned processes
- PID = 2 → **KTHREADD**: **Kernel thread daemon**

- **ZOMBIE PROCESS**: The process has been halted, is dead, but it still has an entry in the process table

PROCESS MANAGEMENT→PROCESS ID

- A NEW process is CREATED based on an existing process, that makes an EXACT COPY of itself in memory
- (the child process will have the same environment as its parent, apart from the PID)
- There are some ways to do that using **system calls:**
  - **system()**
  - **fork()**
  - **exec()**

- **FOREGROUND PROCESS/INTERACTIVE PROCESS**:

They have the control of the terminal

They are controlled by the user (that decides to begin the process, and that can stop/kill the process CTRL+Z, CTRL+C)

They are not part of the system function/services

- **BACKGROUND PROCESS/NON-AUTOMATIC PROCESS**:

They are not connected to a terminal (at that time)

They do not expect any user input

- **DAEMON**

Special type of background process

That can be started with system startup, and used to be kept running forever as a **service**

They are started as **system tasks**

They can be controlled via the **init** process

They can be stopped/restarted

PROCESS MANAGEMENT→RUNNING PROCESSES

- All running processes are mapped according to their PID on **/proc**
  - /proc/[PID]/cmdline: the command that started the process

```
devasc@LJG:~$ more /proc/2225/cmdline
bash
devasc@LJG:~$
```

  - /proc/[PID]/environ: environment variables that affect the process
  - /proc/[PID]/status: information about a process including (run state, memory usage, …)

```
devasc@LJG:~$ ls /proc/2225/
arch_status      environ     mountinfo      personality   statm
attr             exe         mounts         projid_map    status
autogroup        fd          mountstats     root          syscall
auxv             fdinfo      net            sched         task
cgroup           gid_map     ns             schedstat     timers
clear_refs       io          numa_maps      sessionid     timerslack_ns
cmdline          limits      oom_adj        setgroups     uid_map
comm             loginuid    oom_score      smaps         wchan
coredump_filter  map_files   oom_score_adj  smaps_rollup
cpuset           maps        pagemap        stack
cwd              mem         patch_state    stat
devasc@LJG:~$
```