# Processes

Unix timeslices: each runnable process gets an allocation of time units.

The allocation does not depend on it being in the active window,

Processes have priority, but each gets some time (except very nice processes).

Your niceness (priority) controls how much CPU you get, but, if no else is running you will.
Example: all other processes are waiting for disk or user input.

# System Load

`uptime` – reports the short, medium and long term number of runable processes. If this is under 1.0 you've got spare CPU. Warning: multiple CPU machines may or may not divide by the number of processors.

`free` – summary statistics about memory

`vmstat` – summary statistics about processes, swap space, memory usage, CPU usage and io. Shows what you are running short on. With no parameters it lists statistics since the last reboot. To get more meaningful information you can use the parameters to specify how many times to run and at what interval; the first report will still be a summary, but the later ones will be snapshot.

`top` – summarizes statistics and lists most consumptive processes

`ps` – look at the running processes. Various options show different subsets of the processes with different amounts of information. For a complete listing use:
`ps auxww`

# More about the ps command

`%CPU`

The percentage of the CPU a process is using.
This is the current CPU percentage, your ps will be very high since it is running now.
*Watch for:* A process taking a large percentage of time.

`TIME`

This is the total amount of CPU that the process has consumed.
Certain processes such as system will accumulate a lot of CPU since they never stop.
*Watch for:* A process which had consumed alot of CPU.

`STAT`

The status of
*Watch for:* A lot of defunct (`Z`) processes, indicating that some server is incorrectly handling forks and waits.
A lot of IO wait processes (`D`), often indicates that some network file system isn't online or a hard drive had gone bad.

`PID`

You need this number to terminate the process

Watch for abnormalities

Example 1: a `COMMAND` of `telnet` with a `TTY` of ?. Telnets and edits should be associated with terminals

Example 2: a compile (`gcc`) with a `TTY` of ?. This will happen on build servers, but probably not on other machines.

# Sample Administration

Administration Task – remove run-away user jobs (without rebooting).

```
sseiden  13638 90.0  0.4 3164   120 ?  R    Sep  6 10935:50 netscape

volper    1464  7.7  1.8  260   508 p5 R    11:58       0:00 ps aux

root       119  0.0  0.0   28     0 ?  S <  Sep  4     38:50  (nfsd)

root         2  0.0  0.0    0     0 ?  D    Sep  4      0:46 pagedaemon
```

The netscape looks suspicious. It should be interactive. It's using 90% of the CPU. It's used 10000 minutes of CPU.

Action – 1) call user, 2) `kill 13638`, 3) `kill -9 13638` 4) `kill -15 13638`

Caution: killing the wrong process could a) upset an innocent user b) lobotomize the operating system

For a multiprocess jam you can kill all processes of a given name:

```
joe  1431  1.0  0.5  260    54 ?  R   10:39      0:50 hwk1

joe  1432  1.1  0.4  260    54 ?  R   10:39      0:48 hwk1

joe  1433  1.0  0.4  261    54 ?  R   10:39      0:50 hwk1

joe  1434  1.2  0.4  261    54 ?  R   10:39      0:46 hwk1
```

```
killall hwk1
```