

Processes

Every running program is called a *process*. A `vim` that you are running, a shell that you are running, a daemon that is waiting for `ssh` connections are examples of processes.

All Unix systems can run several processes simultaneously.

Listing processes - `ps`

There are two versions of `ps` on babbage — `/usr/ucb/ps` and `/bin/ps` (also installed as `/usr/bin/ps`). Most important `/usr/ucb/ps` options:

- `-a` include processes owned by others
- `-l` long listing
- `-u` user-oriented listing
- `-w` wide output (132 columns rather than 80)
- `-ww` arbitrarily wide output
- `-x` include processes with no controlling terminal
- `nn` list information on process `nn`

Creating new processes

`fork` creates a duplicate (a *child*) of a current process. `exec` replaces the existing process with a new one (the PID stays the same).

Normal execution of a command from shell: `fork-exec-wait`.

`command&` executes the `command` in background: `fork-exec`.

Suspend key (usually `Ctrl-Z`, changed by `stty` utility) suspends a running process.

`jobs` lists the suspended and background jobs (`-l` option gives process IDs), `fg %nn` brings job `nn` into foreground and `bg %nn` brings job `nn` into background (useful for suspended jobs).

`kill -signal %nn` sends a `signal` to job `nn` and `kill -signal pid` send a `signal` to process `pid`.

Useful signals: `HUP` (hang-up), `KILL` and `TERM` (terminate). By default, `kill` sends a `TERM` signal.

Shell startup

bash reads:

at login: `~/.bash_profile`, or else `~/.bash_login`, or else `~/.profile`

when interactive: `~/.bashrc`

non-interactive: `$ENV`

tcsh reads: `~/.tcshrc`, or else `~/.cshrc`

and at login: `~/.login`

Interesting variables

Environment: `PATH`, `TERM`

bash: `PS1`, `HISTSIZE`

tcsh: `prompt`, `rprompt`, `history`, `savehist`, `correct`, `autologout`

Aliases

	bash	tcsh
List all aliases	<code>alias</code>	<code>alias</code>
List a single alias	<code>alias <i>name</i></code>	<code>alias <i>name</i></code>
Set an alias	<code>alias <i>name=value</i></code>	<code>alias <i>name value</i></code>
Delete an alias	<code>unalias</code>	<code>unalias</code>

Passing arguments into an alias (tcsh)

`!n` *n*th argument
`!^` same as !1
`!*` all arguments
`!$` the last argument
`!m-n` range of arguments