LPIC-1 Study Group

2 Managing Software

R. Scott Granneman

scott@granneman.com
www.granneman.com

Last updated 20120826

This presentation
is based on
Roderick W. Smith’s
LPIC-1: Linux Professional Institute Certification Study Guide,
2nd edition

That said,
there are many
additions, subtractions, & changes

Introduction

Package Concepts

Windows installation files

```
setup.exe
install.msi
```

OS X installation files

```
foo.pkg
bar.dmg
```

Linux installation files are *packages*

`foo.rpm`

RPM-based distros

`bar.deb`

Debian-based distros

Just *having* a package
isn’t enough

You have to somehow *manage*
the packages

Linux package managers

```
rpm
```

```
rpm -i foo.rpm
```

RPM-based distros

```
dpkg
```

```
dpkg -i bar.deb
```

Debian-based distros

Package managers
make it easy to
install, upgrade, uninstall, & query
packages,
create packages from source code,
& keep all that info
in a database

Another important thing
package managers do:

track *dependencies*

Some packages have dependencies:
they require other packages
in order to be installed or run

In order to install `foo`,
you must first find & install `bar`

Package managers are great,
but you still have to do things manually

Find packages

Download packages

Resolve dependencies

What we need is ... *automation!*

Automated package managers sit on top of package managers to alleviate manual tasks

`YUM`

RPM-based distros

`APT`

Debian-based distros

To find software,
YUM & APT
look in *repositories*
of software online

Repos are like online stores

Your distro knows about
certain repos by default,
but you can tell YUM or APT
about more

GUIs for
Automated Package Managers

Yumex (YUM Extender)
RPM-based distros

Synaptic
Debian-based distros

	RPM-based	Debian-based
GUI	Yumex ↑	Synaptic ↑
Automated Package Managers	YUM ↑	APT ↑
Package Managers	rpm ↑	dpkg ↑
Packages	foo.rpm	bar.deb

You can install & use RPM
on a Debian-based system,
& vice-versa

Don’t mix & match, though

`dpkg` & `rpm`
use completely different databases

RPM

RPM Distributions & Conventions

RPM was developed by Red Hat

Originally stood for
Red Hat Package Manager

Now stands for
RPM Package Manager

Used by many Linux distros,
even those not based on Red Hat

RPM packages
have certain naming conventions

`packagename-a.b.c-x.arch.rpm`

`packagename`

Name of package:

`samba` or `ghostscript` or `libcups2`

`a.b.c`

Version number: `1.5.3` or `3.0.25b`

`x`

Build or release number: `1` or `5c`

Minor changes
made by package maintainer,
not original programmer

packagename-a.b.c-x.arch.rpm

arch

Architecture: i386 or ppc or x86_64

All source RPMs use src

Example:

samba-3.0.25b-4.5mdv2008.0.x86_64.rpm

samba-|3.0.25b-|4.5mdv2008.0.|
x86_64.rpm

These are just conventions

You could just call it `foo.rpm`

Not very communicative, thoug

Compatability problems between RPMs on different distros

✓ Different versions of RPM

✓ Unmet dependencies

✓ Different names for packages & dependencies

✓ Slightly different files

✓ Distribution-specific scripts or config files

Safest to use RPMs
made for your distro,
but you can always try other RPMs,
& they'll often work

rpm

```
rpm [operation] [options] package
```

`[operation]` is what you want to do: e.g., install, upgrade, uninstall, query

`[options]` change operation: e.g., force, be verbose, test

Options change depending on the operation

```
rpm -ihv foo.rpm
```

`-i`: Install

`-h`: Show hashmarks #####

`-v`: Be verbose

```
rpm -Uhv foo.rpm
```

`-U`: Upgrade if installed,
or install if not installed

```
rpm -e foo.rpm
```

Uninstall (or erase) a package

Other operations

`-F` or `--freshen`

Upgrade package
if earlier version already exists

`-q`

Query a package

`-V` or `--verify`

Verify a package

`--rebuilddb`

Rebuild RPM database

Options		Operations					
		-i	-U	-F	-e	-q	-V
Display hashmarks # to indicate progress	-h --hash	✓	✓	✓			
With -h, be verbose	-v	✓	✓	✓			
Do no dependency checks	--nodeps	✓	✓	✓	✓		
Dry run, but don’t actually install	--test	✓	✓	✓			
Install to different directory	--prefix	✓	✓	✓			
Query or verify all packages	-a --all					✓	✓
Query or verify package that owns specified file	-f --file					✓	✓
Query uninstalled package	-p					✓	
Display package info	-i					✓	
Display packages on which the package depends	-R --requires					✓	
Display files contained in package	-l --list					✓	

Confusing things about `rpm`

`-i` is both an operator & an option

Operators & options are combined

```
rpm -Uhv
```

`-U` is an operator

`-h` & `-v` are options

Extracting Data from RPMs

Extract data
without actually installing

RPM files are actually
modified cpio archives

`rpm2cpio`

Converts RPM to cpio

```
rpm2cpio foo.rpm > foo.cpio |
cpio -i --make-directories
```

Extracts archive
& creates directories

Always do all this
in a new directory you created,
to avoid spewing files

Could also use `alien`
(covered later)

YUM

http://linux.duke.edu/projects/yum/

YUM was developed|
by Yellow Dog Linux
(*Y*ellow Dog *U*pdate *M*anager)
but it’s been adopted
by Red Hat & many other distros

Not all RPM-based distros use YUM,
like SUSE & Mandriva

```
yum [options] [command] [package]
```

```
yum install foo
```

Install package & dependencies

```
yum remove foo
yum erase foo
```

Delete package & dependencies

```
yum check-update
```

Check to see if updates are available
& list them if they are

```
yum update foo
```

Update package foo to latest version

```
yum update
```

Update *all* packages to latest version

```
yum upgrade
```

Upgrade the distro safely

```
yum list foo
```

Display info about foo

```
yum provides foo
yum whatprovides foo
```

Display info about packages providing the foo program or feature

```
yum search foo
```

Search package names, summaries, & more for foo

```
yum info foo
```

Display info about foo

```
yum clean
```

Clean up YUM cache directory at `/path/to/yum/cache`

```
yum resolvedep foo
```

Display packages matching dependency on foo

```
yum deplist foo
```

Display dependencies of foo

```
yum localinstall foo.rpm bar.rpm
```

Install local RPM files,
using YUM to resolve dependencies

```
yum localupdate foo.rpm bar.rpm
```

Update system
using only local RPM files,
using YUM to resolve dependencies

```
yum shell
```

Enter YUM shell mode,
so you can enter
multiple YUM commands

```
yumdownloader foo
```

Download latest version of foo
but don’t install it

RPM & YUM Config

RPM config

`/usr/lib/rpm/rpmrc`

Main RPM config file,
but don’t edit it

Use these:

`/etc/rmprc`

Global changes for all users

`~/.rpmrc`

Personal changes

Main reasons
to customize RPM config
are if you're
converting source RPMs
into binary RPMs:

Pass appropriate compiler options

Build for correct architecture

Pass appropriate compiler options
to set architecture (CPU) optimizations
when you convert
source RPM into binary RPM

For example:

```
optflags: athlon -02 -g -march=i686
```

Pass the `-02 -g -march=i686` options
when building on `athlon`

Build for correct architecture

Default `rpmrc` files include `buildarchtranslate` lines that cause `rpmbuild` to use one set of optimizations for a while family of CPUs

```
buildarchtranslate: athlon: i386
buildarchtranslate: i686: i386
buildarchtranslate: i586: i386
buildarchtranslate: i486: i386
buildarchtranslate: i386: i386
```

Guarantees portability
at the expense of
customization for your CPU

To customize:

```
buildarchtranslate: i686: i686
```

YUM config

`/etc/yum.conf`

Basic options

`/etc/yum.repos.d/`

Several file,
each describing a YUM repo

You can manually add files
pointing to repos,
or just download an RPM
that contains repo info & install it

Debian

Debian Distributions & Conventions

Debian packages
originated with Debian (duh)

Now used by Ubuntu
& many other distros

Naming conventions

`apport_2.0.1-0ubuntu12_all.deb`

`udev_175-0ubuntu9.1_amd64.deb`

`amd64`: 64-bit

`all`: CPU-independent

Use dpkg
for one or a few packages

Use APT
to manage several packages
or the system

dpkg

```
dpkg options action foo.deb bar.deb
```

dpkg actions

`-i` `--install`	Install package
`--configure`	Reconfigure installed package
`-r` `--remove`	Remove package, leaving config files
`-P` `--purge`	Remove package & config files

`-i` expects full package filename

(`foo-1.0-0_1.0-0ubuntu9.1_all.deb`)

All others are fine with `foo`

dpkg actions

`-p` `--print-avail`	Display info about installed package
`-I` `--info`	Display info about uninstalled package

`-I` expects full package filename

(`foo-1.0-0_1.0-0ubuntu9.1_all.deb`)

All others are fine with `foo`

dpkg actions

`-l` *pattern* `--list` *pattern*	List all installed packages matching *pattern*
`-L` `--listfiles`	List installed files for package
`-S` *pattern* `--search` *pattern*	List packages owning files matching *pattern*
`-C` `--audit`	Search for partially installed packages

dpkg options

--force-*things*	Force actions to be taken
--no-act	Check for dependencies, conflicts, & problems; don’t actually install/remove
-G	Don’t install if new version is already installed
-E --skip-same-version	Don’t install if same version is already installed

apt-cache

```
apt-cache
```

Provide info about
Debian package database
(the *package cache*)

```
apt-cache search foo
```

Search for package

```
apt-cache showpkg foo
```

Display info about the package

```
apt-cache stats
```

View statistics
about the package cache

```
apt-cache unmet
```

Find unmet dependencies

```
apt-cache depends foo
```

View package's dependencies

```
apt-cache pkgnames
```

Show all installed packages

apt-get

Full-featured package manager

Uses list of repositories
in `/etc/apt/sources.list`
& `/etc/apt/sources.list.d`

```
$ cat /etc/apt/sources.list

## Note, this file is written by cloud-init on first boot of an instance
## modifications made here will not survive a re-bundle.
## if you wish to make changes you can:
## a.) add 'apt_preserve_sources_list: true' to /etc/cloud/cloud.cfg
##     or do the same in user-data
## b.) add sources in /etc/apt/sources.list.d
## c.) make changes to template file /etc/cloud/templates/sources.list.tmpl

# See http://help.ubuntu.com/community/UpgradeNotes for how to
# upgrade to newer versions of the distribution.
deb http://us-east-1.ec2.archive.ubuntu.com/ubuntu/ precise main
deb-src http://us-east-1.ec2.archive.ubuntu.com/ubuntu/ precise main

## Major bug fix updates produced after the final release of the
## distribution.
deb http://us-east-1.ec2.archive.ubuntu.com/ubuntu/ precise-updates main
deb-src http://us-east-1.ec2.archive.ubuntu.com/ubuntu/ precise-updates main
```

```
apt-get options command package
```

```
apt-get update
```

Get updated info
about packages in repository

```
apt-get upgrade
```

Upgrade all installed packages

```
apt-get update && apt-get upgrade
```

```
apt-get dist-upgrade
```

Upgrade to a new distro version

```
apt-get install foo bar
```

Install package(s)

```
apt-get remove foo bar
```

Remove package(s)

```
apt-get source foo bar
```

Install source packages

`apt-get check`

Check database
for consisitency & broken installs

`apt-get clean`

Remove installed packages from
`/var/cache/apt/archives`

`apt-get autoclean`

Removed installed packages from
`/var/cache/apt/archives`
that can no longer be downloaded

`apt-get` options

`-d` or `--download-only`

Download but don't install

`-f` or `--fix-broken`

Fix dependency problems

`-s` or `--simulate`

or `--dry-run` or `--no-act`

Simulate installation or removal

`-y` or `--yes` or `--assume-yes`

Answer yes to any prompts

`dselect`,
`aptitude`,
& Synaptic

`dselect`

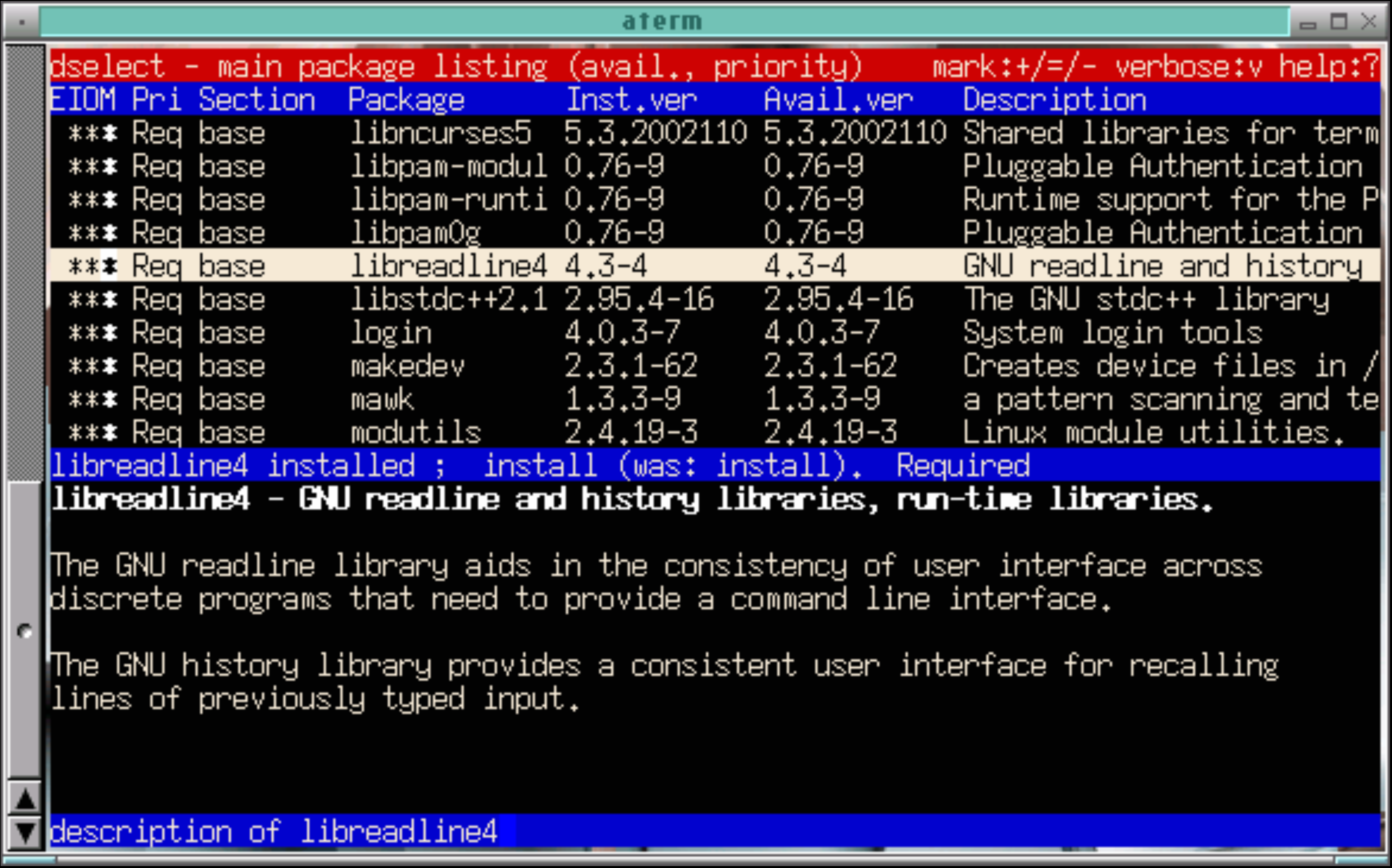
“A high-level interface for managing the installation & removal of Debian software packages.

Many users find `dselect` intimidating & new users may prefer to use apt-based user interfaces.”

Written in the 1990s

According to Wikipedia:

“`dselect` has
a text-mode user interface,
a set of key bindings
that is generally considered
to be fairly non-intuitive,
& its dependency resolution mechanism
is suboptimal.”

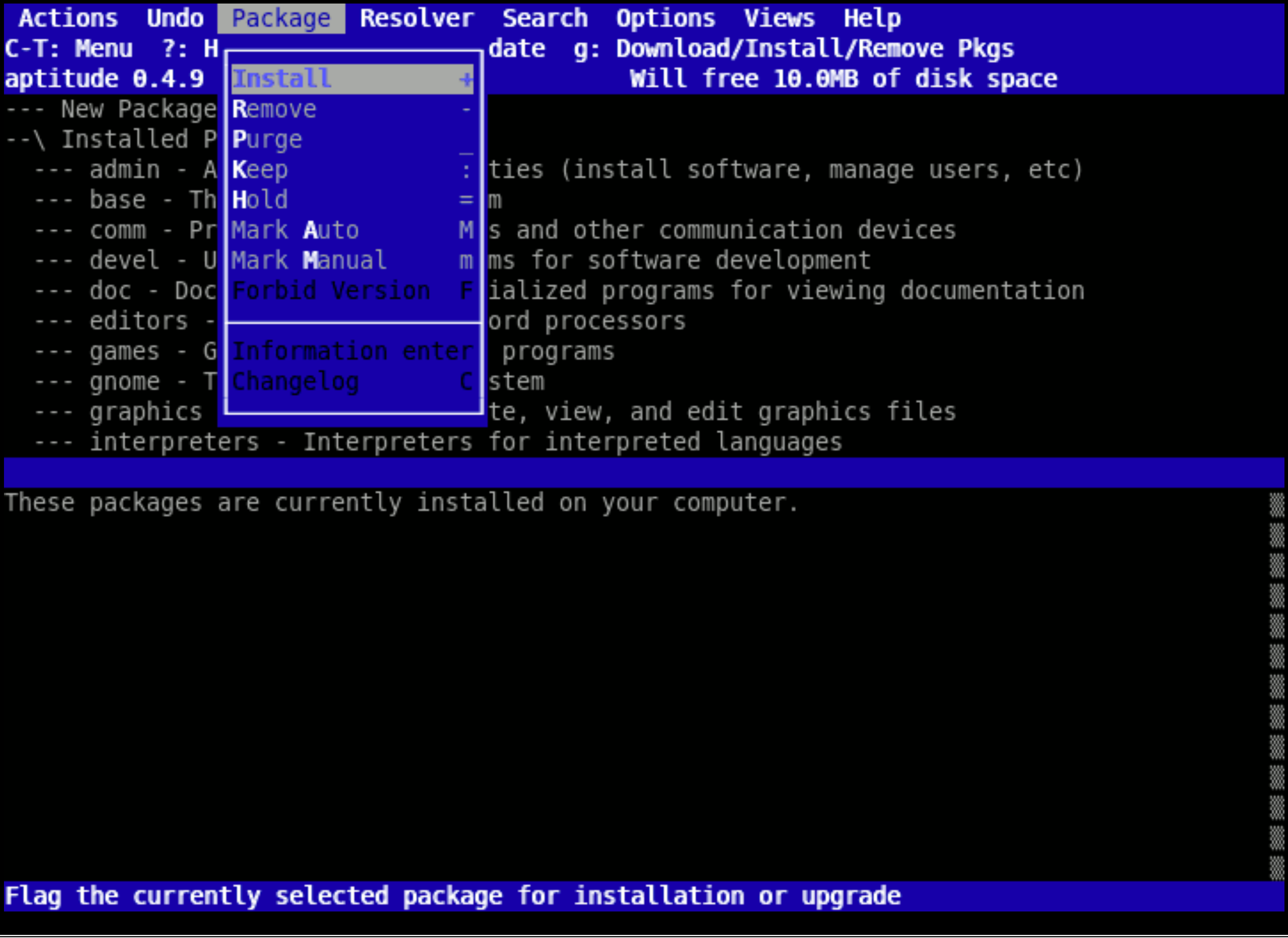


```
dselect - main package listing (avail., priority)    mark:+/=/- verbose:v help:?
EIOM Pri Section  Package      Inst.ver     Avail.ver    Description
 *** Req base     libncurses5  5.3.2002110 5.3.2002110 Shared libraries for term
 *** Req base     libpam-modul 0.76-9       0.76-9       Pluggable Authentication
 *** Req base     libpam-runti 0.76-9       0.76-9       Runtime support for the P
 *** Req base     libpam0g     0.76-9       0.76-9       Pluggable Authentication
 *** Req base     libreadline4 4.3-4        4.3-4        GNU readline and history
 *** Req base     libstdc++2.1 2.95.4-16    2.95.4-16    The GNU stdc++ library
 *** Req base     login        4.0.3-7      4.0.3-7      System login tools
 *** Req base     makedev      2.3.1-62     2.3.1-62     Creates device files in /
 *** Req base     mawk         1.3.3-9      1.3.3-9      a pattern scanning and te
 *** Req base     modutils     2.4.19-3     2.4.19-3     Linux module utilities.
libreadline4 installed ;  install (was: install).  Required
libreadline4 - GNU readline and history libraries, run-time libraries.

The GNU readline library aids in the consistency of user interface across
discrete programs that need to provide a command line interface.

The GNU history library provides a consistent user interface for recalling
lines of previously typed input.



description of libreadline4
```

`aptitude`

Package manager with both
text-mode interactive UI (like `dselect`)
& command line interface

In interactive mode, it's easier than dselect because aptitude adds menus accessed by pressing Ctrl+t



```
 Actions  Undo  Package  Resolver  Search  Options  Views  Help
C-T: Menu  ?: H               date   g: Download/Install/Remove Pkgs
aptitude 0.4.9  Install        +           Will free 10.0MB of disk space
--- New Package Remove         -
--\ Installed P Purge          _
  --- admin - A Keep           : ties (install software, manage users, etc)
  --- base - Th Hold           = m
  --- comm - Pr Mark Auto      M s and other communication devices
  --- devel - U Mark Manual    m ms for software development
  --- doc - Doc Forbid Version F ializzed programs for viewing documentation
  --- editors -                  ord processors
  --- games - G Information enter  programs
  --- gnome - T Changelog      C stem
  --- graphics                   te, view, and edit graphics files
  --- interpreters - Interpreters for interpreted languages

These packages are currently installed on your computer.






Flag the currently selected package for installation or upgrade
```

```
aptitude search foo
```

Search repositories for package foo

```
aptitude update
```

Update package lists

```
aptitude install foo
```

Install package foo

```
aptitude remove foo
```

Remove package foo

Upgrade *all* installed packages

```
aptitude full-upgrade
```

More likely to work, but less safe

```
aptitude safe-upgrade
```

More conservative & safer,
but may fail

```
aptitude autoclean
```

Remove downloaded packages that are no longer available, but keeping others

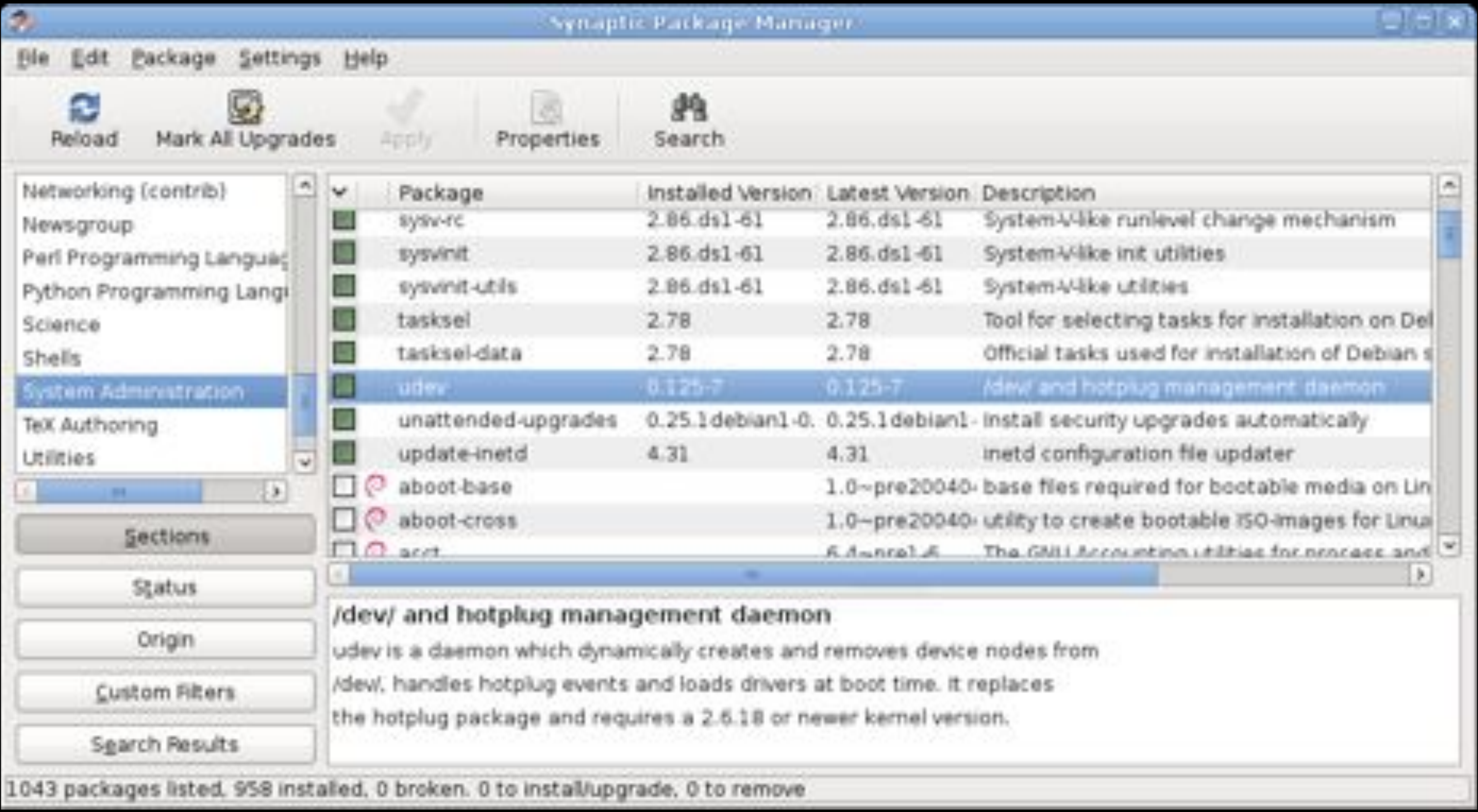
```
aptitude clean
```

Remove *all* downloaded packages, freeing space on your computer

```
aptitude help
```

Show help

Synaptic
GUI interface to APT



		Package	Installed Version	Latest Version	Description
🟩		sysv-rc	2.86.ds1-61	2.86.ds1-61	System-V-like runlevel change mechanism
🟩		sysvinit	2.86.ds1-61	2.86.ds1-61	System-V-like init utilities
🟩		sysvinit-utils	2.86.ds1-61	2.86.ds1-61	System-V-like utilities
🟩		tasksel	2.78	2.78	Tool for selecting tasks for installation on Del
🟩		tasksel-data	2.78	2.78	Official tasks used for installation of Debian s
🟩		udev	0.125-7	0.125-7	/dev/ and hotplug management daemon
🟩		unattended-upgrades	0.25.1debian1-0.	0.25.1debian1-	install security upgrades automatically
🟩		update-inetd	4.31	4.31	inetd configuration file updater
□		aboot-base		1.0~pre20040	base files required for bootable media on Lin
□		aboot-cross		1.0~pre20040	utility to create bootable ISO-images for Linu
□		acct		6.4~pre1-6	The GNU Accounting utilities for process and

/dev/ and hotplug management daemon

udev is a daemon which dynamically creates and removes device nodes from /dev/, handles hotplug events and loads drivers at boot time. It replaces the hotplug package and requires a 2.6.18 or newer kernel version.

1043 packages listed, 958 installed, 0 broken. 0 to install/upgrade, 0 to remove

Reconfiguring Packages

When you install a Debian package, you're sometimes asked configuration questions

To re-configure later, use

```
dpkg-reconfigure foo
```

Debian Compared

Debian	RPM
Source packages are multiple files (source + patch + dsc)	Source packages are a single file
Source packages support 1 patch file	Source packages support >1 patch files
Debian packages more compatible across distros	RPM packages less compatible across distros

The author claims that
“it can be harder
to locate Debian packages
than RPM packages
for some exotic programs”

I would contend
that the opposite could be argued

Configuring Debian Package Tools

Most of the time,
you don’t need to change
dpkg & APT’s defaults

Config files for `dpkg`

```
/etc/dpkg/dpkg.cfg
~/.dpkg.cfg
```

Config files for APT are in `/etc/apt`

`apt.conf`

OR

`apt.conf.d/`

APT & `dselect` options

`sources.list`

List of repositories

Better to use `sources.list.d/`

```
$ pwd
/etc/apt
$ ls sources.list.d
alestic-ppa-precise.list
$ cat sources.list.d/alestic-ppa-
precise.list
deb http://ppa.launchpad.net/↩
   alestic/ppa/ubuntu precise main
deb-src http://ppa.launchpad.net/↩
   alestic/ppa/ubuntu precise main
```

`/var/lib/dpkg`

Lists of
available & installed packages

`/var/cache/apt`

Downloaded & installed packages

Converting Between Formats

`alien`

Convert RPM to `dpkg`,
& vice-versa

Can also convert to & from tarballs

Gotta have `dpkg` & RPM installed

Not always perfect,
but worth a try

Convert between formats

```
alien --to-rpm foo.deb
alien --to-deb foo.rpm
alien --to-tgz foo.rpm
alien --to-deb --install foo.rpm
```

Convert to `dpkg` & install
so APT records info

If you convert or install
from a tarball,
keep in mind that files are installed
starting from /

You may need to untar,
move files around,
re-tar,
& then run `alien`

Dependencies & Conflicts

Sometimes, you’ll run into problems installing packages

You are far less likely
to have problems
if you stick to APT & YUM

Real & Imagined Problems

Missing libraries
or support programs
(QT, GTK, X.org)

Incompatible libraries
or support programs

Duplicate files or features

Mismatched names

Workarounds

Ways to fix the problem:

Forcing

Upgrading or replacing

Rebuilding

Locate another version

Forcing

Install anyway & ignore issues

Be careful!

```
rpm -i foo.rpm --nodeps
```

Install & ignore failed dependencies

```
rpm -i foo.rpm --force
```

Install & ignore errors

```
dpkg --ignore-depends=bar -i foo.deb
```

Ignore dependency checking
& only warn about conflicts

```
dpkg --force-depends -i foo.deb
```

Turn dependency problems
into warnings

```
dpkg --force-conflicts -i foo.deb
```

Install & ignore conflicts

Upgrading or Replacing

The “correct” way to fix problems

Turns into a problem
when you’re running distro A
& you upgrade a package
built for distro B

Rebuilding

When package was built,
certain libraries & support files
were assumed
that your systems lacks

Solution:
rebuild package from source
so it uses
your libraries & support files

```
rpmbuild --rebuild foo.src.rpm
```

Results in new RPM in
/usr/src/*distname*/RPMS/*arch*

Gotta get hold
of the source RPM first!

```
apt-get source foo
```

Download source to foo

```
apt-get build-dep foo
```

Get & install packages required to rebuild foo

```
cd foo
```

```
debuild -us -uc
```

Rebuild foo binary package without signing the .changes file (since you’re not the developer)

`foo.deb` is in parent directory

Locating Another Version

Get a package that’s
newer, older, or
built for different distro

Of course,
you might really need the version
that doesn’t work!

Good places to search for packages

RPM Find
www.rpmfind.net

Fresh RPMs
freshrpms.net

Debian Packages
www.debian.org/distrib/packages

Startup Script Problems

Startup scripts
may not alway work
on different distros

Workarounds

Modifying existing startup script

Writing a new script

Starting the server
through a local startup script like
`/etc/rc.d/rc.local`
or `/etc/rc.d/boot.local`

Managing Shared Libraries

Library Principles

Libraries provide
commonly used code fragments

Helps developers
avoid rebuilding the wheel

Most programs
don’t incorporate libraries
(bloated! slow!)

Instead, they reference
the *shared* (or *dynamic*) library files

Linux names them
`foo.so` or `foo.so.1`

Windows calls these
DLLs (Dynamic Link Libraries)

Sidenote

Linux also uses
static libraries:
code which is linked with,
& is incorporated into,
the program

`foo.a`

Windows calls these `.lib` files

Problems with shared libraries

Degrade program load time
if not already in use elsewhere

Changes to a library
can break programs

Programs need to know
where libraries are

Lots of libraries to manage

Problematic libraries
can break your system

Overall,
the benefits of shared libraries
outweigh the risks

Locating Library Files

Biggest admin challenge
with shared libraries:
making sure programs can find them

Programs can point to libraries
by name (`libc.so.6`)

or path (`/lib/libc.so.6`)

Library path provides programs
with a list of directories
in which to search for libraries

Setting the Library Path Systemwide

`/etc/ld.so.conf`

Sets library path systemwide

Usually never needs to be changed unless you install a library manually in an unusual location

After changing `ld.so.conf`,

use `ldconfig` to update system

(coming up!)

```
$ cat /etc/ld.so.conf
include /etc/ld.so.conf.d/*.conf
$ ls /etc/ld.so.conf.d
libc.conf  x86_64-linux-gnu.conf
$ cat /etc/ld.so.conf.d/*
  # libc default configuration
  /usr/local/lib

  # Multiarch support
  /lib/x86_64-linux-gnu
  /usr/lib/x86_64-linux-gnu
```

Trusted library directories
`/lib` & `/usr/lib`
are always in the library path,
even though they're not
in `ld.so.conf`

Temporarily Changing the Path

Testing a new library?

Install shared libraries
& then set `LD_LIBRARY_PATH`
environment variable

```
export LD_LIBRARY_PATH=/path/to/lib
```

Added to start of search path

To set permanently,
edit your shell startup scripts
or edit `/etc/ls.so.conf`

Correcting Problems

Error?

```
$ gimp
gimp: error while loading shared
libraries: libXinerama.so.1:
cannot open shared object file: No
such file or directory
```

Is the library installed?

If not, install it

If it is,
add directory to
`LD_LIBRARY_PATH`
or `/etc/ld.so.conf`

Is path hard-coded into program?

(Stupid developer)

Create a symbolic (or soft) link
from actual location
to location program expects

Then run `ldconfig`

Library Management Commands

`ldd`

Display program’s
shared library dependencies

`ldconfig`

Update caches & links
used by system
for locating libraries
by re-reading `/etc/ld.so.conf`

`ldd`

Display program’s shared library dependencies

```
$  ldd /usr/bin/htop
   linux-vdso.so.1 =>  (0x00007fff7d392000)
   libncursesw.so.5 => /lib/x86_64-linux-gnu/libncursesw.so.5
(0x00007fcbf9365000)
   libtinfo.so.5 => /lib/x86_64-linux-gnu/libtinfo.so.5 (0x00007fcbf913e000)
   libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007fcbf8e43000)
   libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fcbf8a86000)
   libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fcbf8882000)
   /lib64/ld-linux-x86-64.so.2 (0x00007fcbf959b000)
```

`ldconfig`

Update caches & links
used by system
for locating libraries
by re-reading `/etc/ld.so.conf`
& rebuilding `/etc/ld.so.cache`

`ldconfig` is run automatically
when updating library packages

```
ldconfig -v
```

Be verbose

```
ldconfig -p
```

Print the current cache to stdout

Managing Processes

Command → Program → Process

Program can spawn
more than one process

Vital that you know
how to manage processes

Identifying
Moving into fore- & background
Killing
Adjusting priorities

The First Process

The 1st process during boot
is always `init`

(short for *initialization*)

Started by the kernel

Assigned PID 1

```
# ps aux
USER PID %CPU %MEM STAT COMMAND
root   1  0.0  0.0 Ss   /sbin/init
```

`uname`

Display info about the system

```
$ uname
Linux
```

`-n` `--nodename`	Network hostname	`adam.websanity.com`
`-s` `--kernel-name`	Kernel name	`Linux`
`-v` `--kernel-version`	Kernel build date & time	`#46-Ubuntu SMP Fri Jul 27 17:23:50 UTC 2012`
`-r` `--kernel-release`	Kernel version number	`3.2.0-29-virtual`
`-m` `--machine`	CPU	`x86_64`
`-p` `--processor`	CPU info	`x86_64`
`-i` `--hardware-platform`	Hardware info	`x86_64`
`-o` `--operating-system`	Operating system	`GNU/Linux`

-a or --all

```
$ uname -a
Linux adam.websanity.com
3.2.0-29-virtual #46-Ubuntu SMP
Fri Jul 27 17:23:50 UTC 2012
x86_64 x86_64 x86_64 GNU/Linux
```

Examining Process Lists

`ps`

Displays *p*rocesses' *s*tatus

Extremely important tool
for monitoring & managing
your Linux box

Unfortunately, also very complicated

```
ps [options]
```

3 styles for [options]

UNIX98

```
-aux
```

BSD

```
aux
```

GNU long

```
--user scott
```

Display all processes with my user ID & on my terminal

```
$ ps
  PID TTY          TIME CMD
 2612 pts/1    00:00:00 bash
 7559 pts/1    00:00:00 ps
```

`TTY`: terminal associated with PID

`TIME`: cumulated CPU time in `[DD-]hh:mm:ss`

`CMD`: executable name

ps aux

List all processes

```
$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY    STAT START   TIME COMMAND
root         1  0.0  0.0  24412  2280 ?      Ss   Aug17   0:01 /sbin/init
root         2  0.0  0.0      0     0 ?      S    Aug17   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?      S    Aug17   1:34 [ksoftirqd/0]
root       192  0.0  0.0  25384  1312 ?      S    Aug17   0:00 mountall --daem
root       248  0.0  0.0  21520  1212 ?      Ss   Aug17   0:00 /sbin/udevd --d
root       522  0.0  0.0  49948  2876 ?      Ss   Aug17   0:07 /usr/sbin/sshd
root       599  0.0  0.0  14496   920 tty4   Ss+  Aug17   0:00 /sbin/getty -8
root       639  0.0  0.0  19104  1040 ?      Ss   Aug17   0:04 cron
mysql      690 17.0  2.0 1909084 360868 ?    Ssl  Aug17 1458:47 /usr/sbin/mysq
root      6628  0.0  0.0  16972  1760 pts/0  S+   23:16   0:00 /usr/bin/man ps
root      6638  0.0  0.0  12456   996 pts/0  S+   23:16   0:00 pager -s
www-data 6665  1.0  0.3 549808 57868 ?      S    23:17   0:03 /usr/sbin/apach
```

ps aux | grep foo

can be very helpful

`--help`

Display help

`-A` or `-e`

Display all processes on the system

`x`

Displays all processes owned by the user running `ps`

Also increases amount of info displayed about each process

`-u` *`user`* or `U` *`user`* or `--User` *`user`*

Display processes owned by *`user`*

User variable may be username (`scott`) or a user ID (`501`)

Change info that `ps` provides

`-f`	Full-format listing, including command arguments
`-l`	Long format
`j`	BSD job control format
`l`	BSD long format
`u`	User-oriented format
`v`	Virtual memory format

There are many others...

View processes as a hierarchy, so you know what spawned what

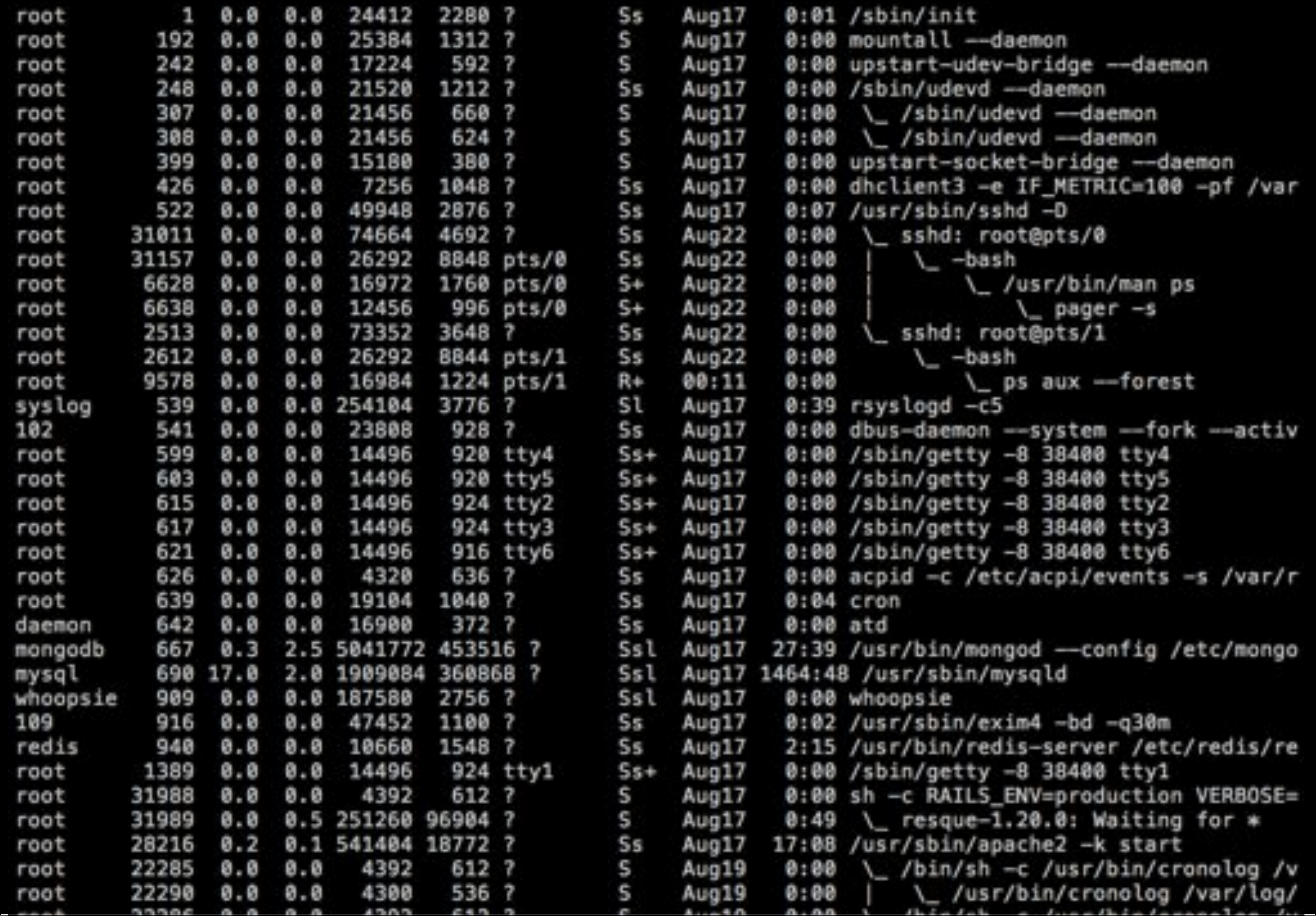
`-H`

Show process hierarchy

`f` or `--forest`

Show process hierarchy using ASCII art (forest)

ps aux --forest



```
root         1  0.0  0.0  24412  2280 ?        Ss   Aug17   0:01 /sbin/init
root       192  0.0  0.0  25384  1312 ?        S    Aug17   0:00 mountall --daemon
root       242  0.0  0.0  17224   592 ?        S    Aug17   0:00 upstart-udev-bridge --daemon
root       248  0.0  0.0  21520  1212 ?        Ss   Aug17   0:00 /sbin/udevd --daemon
root       307  0.0  0.0  21456   660 ?        S    Aug17   0:00  \_ /sbin/udevd --daemon
root       308  0.0  0.0  21456   624 ?        S    Aug17   0:00  \_ /sbin/udevd --daemon
root       399  0.0  0.0  15180   380 ?        S    Aug17   0:00 upstart-socket-bridge --daemon
root       426  0.0  0.0   7256  1048 ?        Ss   Aug17   0:00 dhclient3 -e IF_METRIC=100 -pf /var
root       522  0.0  0.0  49948  2876 ?        Ss   Aug17   0:07 /usr/sbin/sshd -D
root     31011  0.0  0.0  74664  4692 ?        Ss   Aug22   0:00  \_ sshd: root@pts/0
root     31157  0.0  0.0  26292  8848 pts/0    Ss   Aug22   0:00  |   \_ -bash
root      6628  0.0  0.0  16972  1760 pts/0    S+   Aug22   0:00  |       \_ /usr/bin/man ps
root      6638  0.0  0.0  12456   996 pts/0    S+   Aug22   0:00  |           \_ pager -s
root      2513  0.0  0.0  73352  3648 ?        Ss   Aug22   0:00  \_ sshd: root@pts/1
root      2612  0.0  0.0  26292  8844 pts/1    Ss   Aug22   0:00      \_ -bash
root      9578  0.0  0.0  16984  1224 pts/1    R+   00:11   0:00          \_ ps aux --forest
syslog     539  0.0  0.0 254104  3776 ?        Sl   Aug17   0:39 rsyslogd -c5
102        541  0.0  0.0  23808   928 ?        Ss   Aug17   0:00 dbus-daemon --system --fork --activ
root       599  0.0  0.0  14496   920 tty4     Ss+  Aug17   0:00 /sbin/getty -8 38400 tty4
root       603  0.0  0.0  14496   920 tty5     Ss+  Aug17   0:00 /sbin/getty -8 38400 tty5
root       615  0.0  0.0  14496   924 tty2     Ss+  Aug17   0:00 /sbin/getty -8 38400 tty2
root       617  0.0  0.0  14496   924 tty3     Ss+  Aug17   0:00 /sbin/getty -8 38400 tty3
root       621  0.0  0.0  14496   916 tty6     Ss+  Aug17   0:00 /sbin/getty -8 38400 tty6
root       626  0.0  0.0   4320   636 ?        Ss   Aug17   0:00 acpid -c /etc/acpi/events -s /var/r
root       639  0.0  0.0  19104  1040 ?        Ss   Aug17   0:04 cron
daemon     642  0.0  0.0  16900   372 ?        Ss   Aug17   0:00 atd
mongodb    667  0.3  2.5 5041772 453516 ?      Ssl  Aug17  27:39 /usr/bin/mongod --config /etc/mongo
mysql      690 17.0  2.0 1909084 360868 ?      Ssl  Aug17 1464:48 /usr/sbin/mysqld
whoopsie   909  0.0  0.0 187580  2756 ?        Ssl  Aug17   0:00 whoopsie
109        916  0.0  0.0  47452  1100 ?        Ss   Aug17   0:02 /usr/sbin/exim4 -bd -q30m
redis      940  0.0  0.0  10660  1548 ?        Ss   Aug17   2:15 /usr/bin/redis-server /etc/redis/re
root      1389  0.0  0.0  14496   924 tty1     Ss+  Aug17   0:00 /sbin/getty -8 38400 tty1
root     31988  0.0  0.0   4392   612 ?        S    Aug17   0:00 sh -c RAILS_ENV=production VERBOSE=
root     31989  0.0  0.5 251260 96904 ?        S    Aug17   0:49  \_ resque-1.20.0: Waiting for *
root     28216  0.2  0.1 541404 18772 ?        Ss   Aug17  17:08 /usr/sbin/apache2 -k start
root     22285  0.0  0.0   4392   612 ?        S    Aug19   0:00  \_ /bin/sh -c /usr/bin/cronolog /v
root     22290  0.0  0.0   4300   536 ?        S    Aug19   0:00  |   \_ /usr/bin/cronolog /var/log/
```

`ps` normally truncates its output
so it fits on your screen

`-w` & `w`

Go *w*ide & do not truncate

Best then to use

```
ps w > ps.txt
```

```
$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY    STAT START   TIME COMMAND
root         1  0.0  0.0  24412  2280 ?      Ss   Aug17   0:01 /sbin/init
root         2  0.0  0.0      0     0 ?      S    Aug17   0:00 [kthreadd]
root       522  0.0  0.0  49948  2876 ?      Ss   Aug17   0:07 /usr/sbin/sshd
root       599  0.0  0.0  14496   920 tty4   Ss+  Aug17   0:00 /sbin/getty -8
root       639  0.0  0.0  19104  1040 ?      Ss   Aug17   0:04 cron
mysql      690 17.0  2.0 1909084 360868 ?    Ssl  Aug17 1458:47 /usr/sbin/mysq
root      6628  0.0  0.0  16972  1760 pts/0  S+   23:16   0:00 /usr/bin/man ps
www-data 6665  1.0  0.3 549808 57868 ?      S    23:17   0:03 /usr/sbin/apach
```

`USER`: User who started the process

`PID`: Number of the process

`%CPU`: Percentage of CPU time
the process uses while `ps` executes

```
$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY    STAT START   TIME COMMAND
root        1  0.0  0.0  24412  2280 ?      Ss   Aug17   0:01 /sbin/init
root        2  0.0  0.0      0     0 ?      S    Aug17   0:00 [kthreadd]
root      522  0.0  0.0  49948  2876 ?      Ss   Aug17   0:07 /usr/sbin/sshd
root      599  0.0  0.0  14496   920 tty4   Ss+  Aug17   0:00 /sbin/getty -8
root      639  0.0  0.0  19104  1040 ?      Ss   Aug17   0:04 cron
mysql     690 17.0  2.0 1909084 360868 ?    Ssl  Aug17 1458:47 /usr/sbin/mysq
root     6628  0.0  0.0  16972  1760 pts/0  S+   23:16   0:00 /usr/bin/man ps
www-data 6665  1.0  0.3 549808 57868 ?      S    23:17   0:03 /usr/sbin/apach
```

`%MEM`: Percentage of memory process uses

`VSZ`: *V*irtual memory *s*i*z*e of the process in KiB (1024-byte units)

`RSS`: Resident Set Size (non-virtual memory used by the program & its data) in KiB

```
$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY    STAT START   TIME COMMAND
root         1  0.0  0.0  24412  2280 ?      Ss   Aug17   0:01 /sbin/init
root         2  0.0  0.0      0     0 ?      S    Aug17   0:00 [kthreadd]
root       522  0.0  0.0  49948  2876 ?      Ss   Aug17   0:07 /usr/sbin/sshd
root       599  0.0  0.0  14496   920 tty4   Ss+  Aug17   0:00 /sbin/getty -8
root       639  0.0  0.0  19104  1040 ?      Ss   Aug17   0:04 cron
mysql      690 17.0  2.0 1909084 360868 ?    Ssl  Aug17 1458:47 /usr/sbin/mysq
root      6628  0.0  0.0  16972  1760 pts/0  S+   23:16   0:00 /usr/bin/man ps
www-data 6665  1.0  0.3 549808 57868 ?      S    23:17   0:03 /usr/sbin/apach
```

`TTY`: *T*ele*ty*pe code

identifying a terminal session

(Not all processes have TTY numbers, like X programs & daemons)

`STAT`: Process State Code

D	Uninterruptible sleep (usually IO)
R	*R*unning or runnable (in run queue)
S	Interruptible *s*leep (waiting for an event to complete)
T	S*t*opped
X	Dead (should never be seen)
Z	*Z*ombie! Dead but not reaped by its parent

If you’re using BSD formats…

`<`	High priority, so not *nice*
`N`	Low priority, so *n*ice
`L`	Pages *l*ocked into memory (for real-time IO)
`l`	Mu*l*ti-threaded
`+`	In foreground process group

```
$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY    STAT START   TIME COMMAND
root         1  0.0  0.0  24412  2280 ?      Ss   Aug17   0:01 /sbin/init
root         2  0.0  0.0      0     0 ?      S    Aug17   0:00 [kthreadd]
root       522  0.0  0.0  49948  2876 ?      Ss   Aug17   0:07 /usr/sbin/sshd
root       599  0.0  0.0  14496   920 tty4   Ss+  Aug17   0:00 /sbin/getty -8
root       639  0.0  0.0  19104  1040 ?      Ss   Aug17   0:04 cron
mysql      690 17.0  2.0 1909084 360868 ?    Ssl  Aug17 1458:47 /usr/sbin/mysq
root      6628  0.0  0.0  16972  1760 pts/0  S+   23:16   0:00 /usr/bin/man ps
www-data 6665  1.0  0.3 549808 57868 ?      S    23:17   0:03 /usr/sbin/apach
```

`START`: Time the command started,
in `HH:MM` format (if <24 hours)
or `MONDD` (if >24 hours)

`TIME`: cumulated CPU time
in `[DD-]hh:mm:ss` format

`COMMAND`: What launched the process

`top`

Display *top* CPU processes
in real time

By default,
processes are sorted by CPU use,
with biggest at the top

```
top - 02:33:32 up 6 days,  1:40,  2 users,  load average: 0.98, 0.92, 1.26
Tasks: 262 total,   1 running, 260 sleeping,   0 stopped,   1 zombie
Cpu(s):  9.0%us,  4.9%sy,  0.0%ni, 84.1%id,  1.6%wa,  0.0%hi,  0.2%si,  0.3%st
Mem:  17489832k total, 16727580k used,   762252k free,   638272k buffers
Swap:        0k total,        0k used,        0k free, 14317712k cached

PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
18232 www-data  20   0  534m  34m  17m S   18  0.2   0:01.46 apache2
17358 root      20   0 51028  11m 4012 S    7  0.1   1:54.84 s3cmd
  690 mysql     20   0 1864m 352m 4992 S    6  2.1   1479:31 mysqld
  170 root      20   0     0    0    0 S    0  0.0   0:39.88 jbd2/xvda1-8
18277 root      20   0 17468 1432  956 R    0  0.0   0:00.07 top
18288 www-data  20   0     0    0    0 Z    0  0.0   0:00.01 apache2 <defunct>
22456 root      20   0  865m 2532 1836 S    0  0.0  14:20.28 PassengerHelper
    1 root      20   0 24412 2280 1244 S    0  0.0   0:01.67 init
    2 root      20   0     0    0    0 S    0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S    0  0.0   1:35.40 ksoftirqd/0
    4 root      20   0     0    0    0 S    0  0.0   0:00.00 kworker/0:0
    5 root      20   0     0    0    0 S    0  0.0   0:00.00 kworker/u:0
    6 root      RT   0     0    0    0 S    0  0.0   0:00.00 migration/0
    7 root      RT   0     0    0    0 S    0  0.0   0:04.11 watchdog/0
    8 root      RT   0     0    0    0 S    0  0.0   0:00.00 migration/1
    9 root      20   0     0    0    0 S    0  0.0   0:00.00 kworker/1:0
```

Change `top` while it's running

`h` `?`	Display help
`k`	Kill process (enter a PID)
`r`	*`renice`*: change a process' priority (enter a PID & a priority number)
`s`	Change display rate (default is 5 seconds)
`M`	Sort by memory usage
`P`	Sort by CPU usage (the default)
`q`	Quit `top`

Options you can pass `top` when you run it

```
top -d 10
```

Change default delay between updates, in seconds

```
top -p 10220 -p 10221 -p 10222
```

Monitor specific PIDs (up to 20)

```
top -n 10
```

Display number of iterations & quit

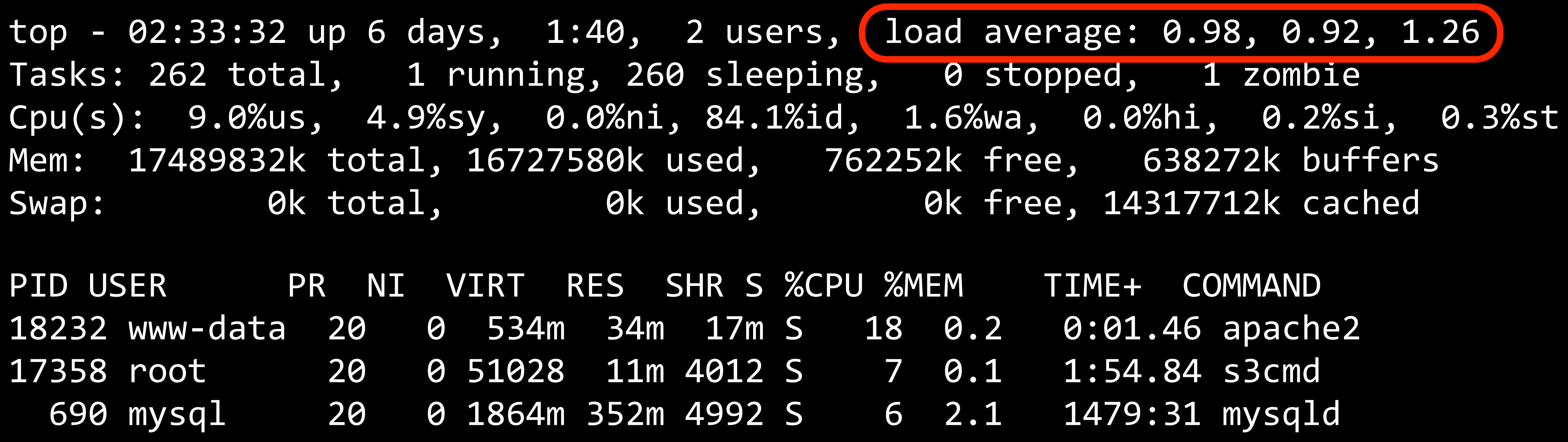
```
top -b > top.txt
```

Run `top` in batch mode,
without updating stdout

Must press `Ctrl-c` to cancel top!

```
top -b -n 5 > top.txt
```

Run `top` in batch mode
for 5 iterations



```
top - 02:33:32 up 6 days,  1:40,  2 users,  load average: 0.98, 0.92, 1.26
Tasks: 262 total,   1 running, 260 sleeping,   0 stopped,   1 zombie
Cpu(s):  9.0%us,  4.9%sy,  0.0%ni, 84.1%id,  1.6%wa,  0.0%hi,  0.2%si,  0.3%st
Mem:  17489832k total, 16727580k used,   762252k free,   638272k buffers
Swap:        0k total,        0k used,        0k free, 14317712k cached

PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
18232 www-data  20   0  534m  34m  17m S   18  0.2   0:01.46 apache2
17358 root      20   0 51028  11m 4012 S    7  0.1   1:54.84 s3cmd
  690 mysql     20   0 1864m 352m 4992 S    6  2.1   1479:31 mysqld
```

Load Average
shows average CPU usage
over 1-, 5-, & 15-minute intervals

0 = idle computer with no tasks

Each process increments by 1

1 CPU

Load average acts as a percentage of system usage

0.98, 0.92, 1.26

0.98: Just about perfect (98% used)

0.92: Just about perfect (92% used)

1.26: Overloaded by 26%,
so .26 processes had to wait

2 CPUs

$$\frac{\text{Load average}}{\text{Number of CPUs}} = \begin{array}{c}\text{Percentage of}\\ \text{system utilization}\end{array}$$

0.98, 0.92, 1.26

0.98/2 = .49
49% used: Twice as fast as needed

0.92/2 = .46
46% used: Twice as fast as needed

1.26/2 = .63
63% used: Still underutilized

4 CPUs

$$\frac{\text{Load average}}{\text{Number of CPUs}} = \begin{array}{c}\text{Percentage of} \\ \text{system utilization}\end{array}$$

1.73, 0.50, 7.98

1.73/4 = .43
43% used: Twice as fast as needed

0.50/4 = .125
13% used: 10x as fast as needed

7.98/4 = 1.99
199% used: Overloaded by 99%,
so 1 process had to wait

`uptime`

Shows how long computer has been running

Also shows load average

```
$ uptime
14:34:03 up 10:43, 4 users,
load average: 0.06, 0.11, 0.09
```

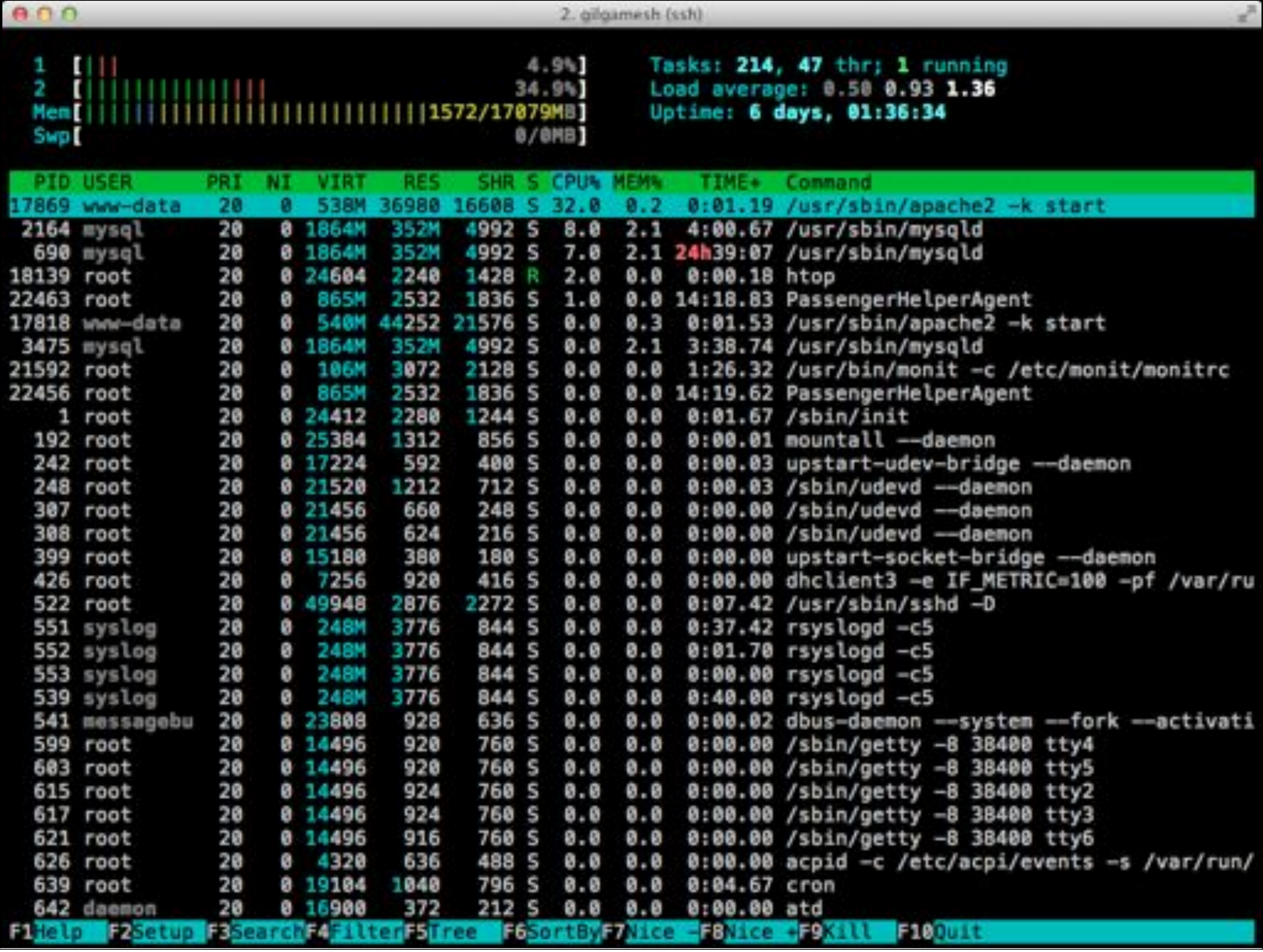
Sidenote:

I prefer htop,
a 3rd party tool that's a better top

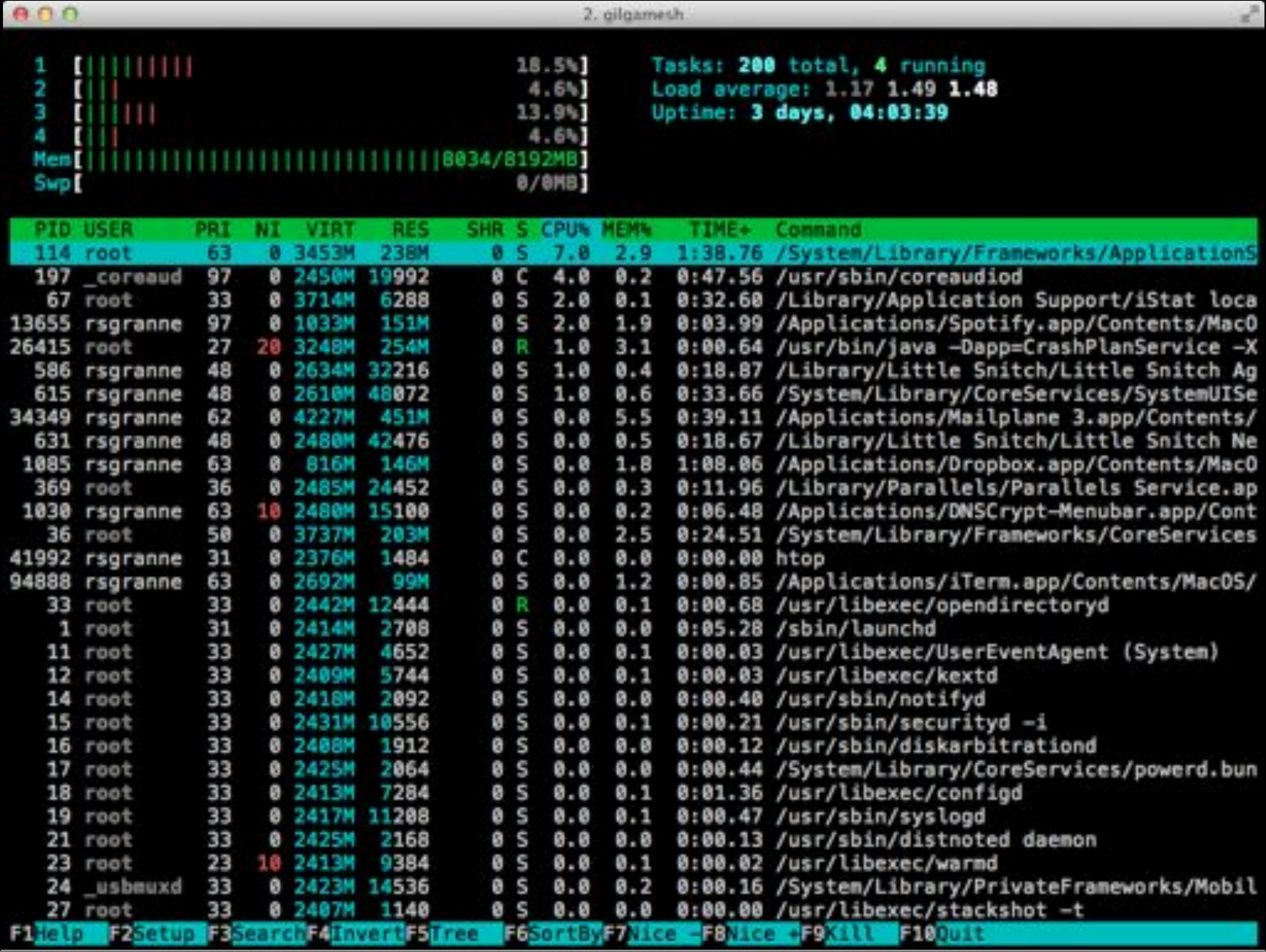
Scroll horizontally & vertically

Faster to start & quicker to use

htop.sourceforge.net



```
  1  [|||                                   4.9%]     Tasks: 214, 47 thr; 1 running
  2  [||||||||||||||||                     34.9%]     Load average: 0.50 0.93 1.36
  Mem[|||||||||||||||||||||||||||||1572/17079MB]     Uptime: 6 days, 01:36:34
  Swp[                                     0/0MB]

  PID USER       PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
17869 www-data    20   0  538M 36980 16608 S 32.0  0.2  0:01.19 /usr/sbin/apache2 -k start
 2164 mysql       20   0 1864M  352M  4992 S  8.0  2.1  4:00.67 /usr/sbin/mysqld
  690 mysql       20   0 1864M  352M  4992 S  7.0  2.1 24h39:07 /usr/sbin/mysqld
18139 root        20   0 24604  2240  1428 R  2.0  0.0  0:00.18 htop
22463 root        20   0  865M  2532  1836 S  1.0  0.0 14:18.83 PassengerHelperAgent
17818 www-data    20   0  540M 44252 21576 S  0.0  0.3  0:01.53 /usr/sbin/apache2 -k start
 3475 mysql       20   0 1864M  352M  4992 S  0.0  2.1  3:38.74 /usr/sbin/mysqld
21592 root        20   0  106M  3072  2128 S  0.0  0.0  1:26.32 /usr/bin/monit -c /etc/monit/monitrc
22456 root        20   0  865M  2532  1836 S  0.0  0.0 14:19.62 PassengerHelperAgent
    1 root        20   0 24412  2280  1244 S  0.0  0.0  0:01.67 /sbin/init
  192 root        20   0 25384  1312   856 S  0.0  0.0  0:00.01 mountall --daemon
  242 root        20   0 17224   592   400 S  0.0  0.0  0:00.03 upstart-udev-bridge --daemon
  248 root        20   0 21520  1212   712 S  0.0  0.0  0:00.03 /sbin/udevd --daemon
  307 root        20   0 21456   660   248 S  0.0  0.0  0:00.00 /sbin/udevd --daemon
  308 root        20   0 21456   624   216 S  0.0  0.0  0:00.00 /sbin/udevd --daemon
  399 root        20   0 15180   380   180 S  0.0  0.0  0:00.00 upstart-socket-bridge --daemon
  426 root        20   0  7256   920   416 S  0.0  0.0  0:00.00 dhclient3 -e IF_METRIC=100 -pf /var/ru
  522 root        20   0 49948  2876  2272 S  0.0  0.0  0:07.42 /usr/sbin/sshd -D
  551 syslog      20   0  248M  3776   844 S  0.0  0.0  0:37.42 rsyslogd -c5
  552 syslog      20   0  248M  3776   844 S  0.0  0.0  0:01.70 rsyslogd -c5
  553 syslog      20   0  248M  3776   844 S  0.0  0.0  0:00.00 rsyslogd -c5
  539 syslog      20   0  248M  3776   844 S  0.0  0.0  0:40.00 rsyslogd -c5
  541 messagebu   20   0 23808   928   636 S  0.0  0.0  0:00.02 dbus-daemon --system --fork --activati
  599 root        20   0 14496   920   760 S  0.0  0.0  0:00.00 /sbin/getty -8 38400 tty4
  603 root        20   0 14496   920   760 S  0.0  0.0  0:00.00 /sbin/getty -8 38400 tty5
  615 root        20   0 14496   924   760 S  0.0  0.0  0:00.00 /sbin/getty -8 38400 tty2
  617 root        20   0 14496   924   760 S  0.0  0.0  0:00.00 /sbin/getty -8 38400 tty3
  621 root        20   0 14496   916   760 S  0.0  0.0  0:00.00 /sbin/getty -8 38400 tty6
  626 root        20   0  4320   636   488 S  0.0  0.0  0:00.00 acpid -c /etc/acpi/events -s /var/run/
  639 root        20   0 19104  1040   796 S  0.0  0.0  0:04.67 cron
  642 daemon      20   0 16900   372   212 S  0.0  0.0  0:00.00 atd
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```



```
  1  [|||||||||                           18.5%]     Tasks: 200 total, 4 running
  2  [|||                                  4.6%]     Load average: 1.17 1.49 1.48
  3  [||||||                              13.9%]     Uptime: 3 days, 04:03:39
  4  [|||                                  4.6%]
  Mem[||||||||||||||||||||||||||||||||8034/8192MB]
  Swp[                                    0/0MB]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
  114 root       63   0 3453M  238M     0 S  7.0  2.9  1:38.76 /System/Library/Frameworks/ApplicationS
  197 _coreaud   97   0 2450M 19992     0 C  4.0  0.2  0:47.56 /usr/sbin/coreaudiod
   67 root       33   0 3714M  6288     0 S  2.0  0.1  0:32.60 /Library/Application Support/iStat loca
13655 rsgranne   97   0 1033M  151M     0 S  2.0  1.9  0:03.99 /Applications/Spotify.app/Contents/MacO
26415 root       27  20 3248M  254M     0 R  1.0  3.1  0:00.64 /usr/bin/java -Dapp=CrashPlanService -X
  586 rsgranne   48   0 2634M 32216     0 S  1.0  0.4  0:18.87 /Library/Little Snitch/Little Snitch Ag
  615 rsgranne   48   0 2610M 48072     0 S  1.0  0.6  0:33.66 /System/Library/CoreServices/SystemUISe
34349 rsgranne   62   0 4227M  451M     0 S  0.0  5.5  0:39.11 /Applications/Mailplane 3.app/Contents/
  631 rsgranne   48   0 2480M 42476     0 S  0.0  0.5  0:18.67 /Library/Little Snitch/Little Snitch Ne
 1085 rsgranne   63   0  816M  146M     0 S  0.0  1.8  1:08.06 /Applications/Dropbox.app/Contents/MacO
  369 root       36   0 2485M 24452     0 S  0.0  0.3  0:11.96 /Library/Parallels/Parallels Service.ap
 1030 rsgranne   63  10 2480M 15100     0 S  0.0  0.2  0:06.48 /Applications/DNSCrypt-Menubar.app/Cont
   36 root       50   0 3737M  203M     0 S  0.0  2.5  0:24.51 /System/Library/Frameworks/CoreServices
41992 rsgranne   31   0 2376M  1484     0 C  0.0  0.0  0:00.00 htop
94888 rsgranne   63   0 2692M   99M     0 S  0.0  1.2  0:00.85 /Applications/iTerm.app/Contents/MacOS/
   33 root       33   0 2442M 12444     0 R  0.0  0.1  0:00.68 /usr/libexec/opendirectoryd
    1 root       31   0 2414M  2708     0 S  0.0  0.0  0:05.28 /sbin/launchd
   11 root       33   0 2427M  4652     0 S  0.0  0.1  0:00.03 /usr/libexec/UserEventAgent (System)
   12 root       33   0 2409M  5744     0 S  0.0  0.1  0:00.03 /usr/libexec/kextd
   14 root       33   0 2418M  2092     0 S  0.0  0.0  0:00.40 /usr/sbin/notifyd
   15 root       33   0 2431M 10556     0 S  0.0  0.1  0:00.21 /usr/sbin/securityd -i
   16 root       33   0 2408M  1912     0 S  0.0  0.0  0:00.12 /usr/sbin/diskarbitrationd
   17 root       33   0 2425M  2064     0 S  0.0  0.0  0:00.44 /System/Library/CoreServices/powerd.bun
   18 root       33   0 2413M  7284     0 S  0.0  0.1  0:01.36 /usr/libexec/configd
   19 root       33   0 2417M 11208     0 S  0.0  0.1  0:00.47 /usr/sbin/syslogd
   21 root       33   0 2425M  2168     0 S  0.0  0.0  0:00.13 /usr/sbin/distnoted daemon
   23 root       23  10 2413M  9384     0 S  0.0  0.1  0:00.02 /usr/libexec/warmd
   24 _usbmuxd   33   0 2423M 14536     0 S  0.0  0.2  0:00.16 /System/Library/PrivateFrameworks/Mobil
   27 root       33   0 2407M  1140     0 S  0.0  0.0  0:00.00 /usr/libexec/stackshot -t
F1Help  F2Setup F3SearchF4InvertF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

`jobs`

Display info about processes associated with current session

List job ID numbers (not the same as PIDs)

Ensure all programs have terminated before shutting down

```
$ jobs -l
+[4] 139  Running  CC - C foo c&
-[3] 465  Stopped  mail alice
 [2] 687  Done(1)  foo.bar&
```

+ identifies default job
for the `fg` or `bg` commands

- identifies job
that would become the new default
if current default job exits

Foreground & Background Processes

Normally,
when you run a program,
it takes over the terminal

What if you need
to run another program?

Ctrl-z

Suspend current program
& go back to terminal prompt

`fg`

Restore suspended program
back to *f*ore*g*round

`fg 3`

Restore numbered job
if several are suspended

```
bg
```

Restore a job to running status
(after pressing Ctrl-z),
but in the background

```
foo &
```

Start a program
& run it in the background

Managing Priorities

Want to prioritize programs' CPU use?

Run CPU-intensive job
so it doesn't bog down system?

Give a job more CPU
because it's more important?

Be *nice*

`nice`

Assign CPU priority to a program

`renice`

Alter CPU priority
of a running program

Options for assigning priority

-*priority*

-n *priority*

--adjustment=*priority*

Priority can range from -20 to 19

Default is 0

```
nice -n 12 foo
```

Start foo with a priority of 12
so it uses more CPU

```
renice priority PID
```

Change priority for PID

```
renice priority -g group
```

Change priority for group

```
renice priority -u user
```

Change priority for user

Or combine options
& change priority
for program, user, &/or group:

```
renice priority PID -g group -u user

renice -5 10010 -g staff -u frank
```

Killing Processes

`kill`

Terminate a process
based on its PID

`killall`

Terminate a process
based on its name

`kill`

Terminate a process
based on its PID

(Get PID from `ps` or `top`)

```
kill -signal PID

kill -1 10110

kill -9 10111

kill -TERM 10112
```

1	`HUP`	Kill interactive programs & daemons reread config files
9	`KILL`	Kill program immediately, without saving
15	`TERM`	Kill program but allow it to close open files (the default)

```
kill -l
```

See full list of signals

`killall`

Terminate a process based on its name

```
killall apache2
```

```
killall vim
```

Use `killall` with process name, & nothing else

```
$ ps aux
www-data 31431 … /usr/sbin/apache2↩
    -k start
www-data 31434 … /usr/sbin/apache2↩
    -k start
www-data 31436 … /usr/sbin/apache2↩
    -k start
$ killall apache2
```

When you log out of a shell session,
the kernel sends programs
the `SIGHUP` signal to terminate

What if you want a program
to continue running
after you log out?

```
$ nohup foobar
```

Tells the program foobar
to run & ignore `SIGHUP` signals

`screen` is another method,
though not covered on the LPIC

```
$ man screen
```

Google *screen tutorial*

Review

Thank you!

Email: scott@granneman.com

Web: www.granneman.com

Publications: www.granneman.com/pubs

Blog: ChainSawOnATireSwing.com

Twitter: scottgranneman

LPIC-1 Study Group

2 Managing Software

R. Scott Granneman

scott@granneman.com
www.granneman.com

Licensing of this work

Questions? Email scott@granneman.com