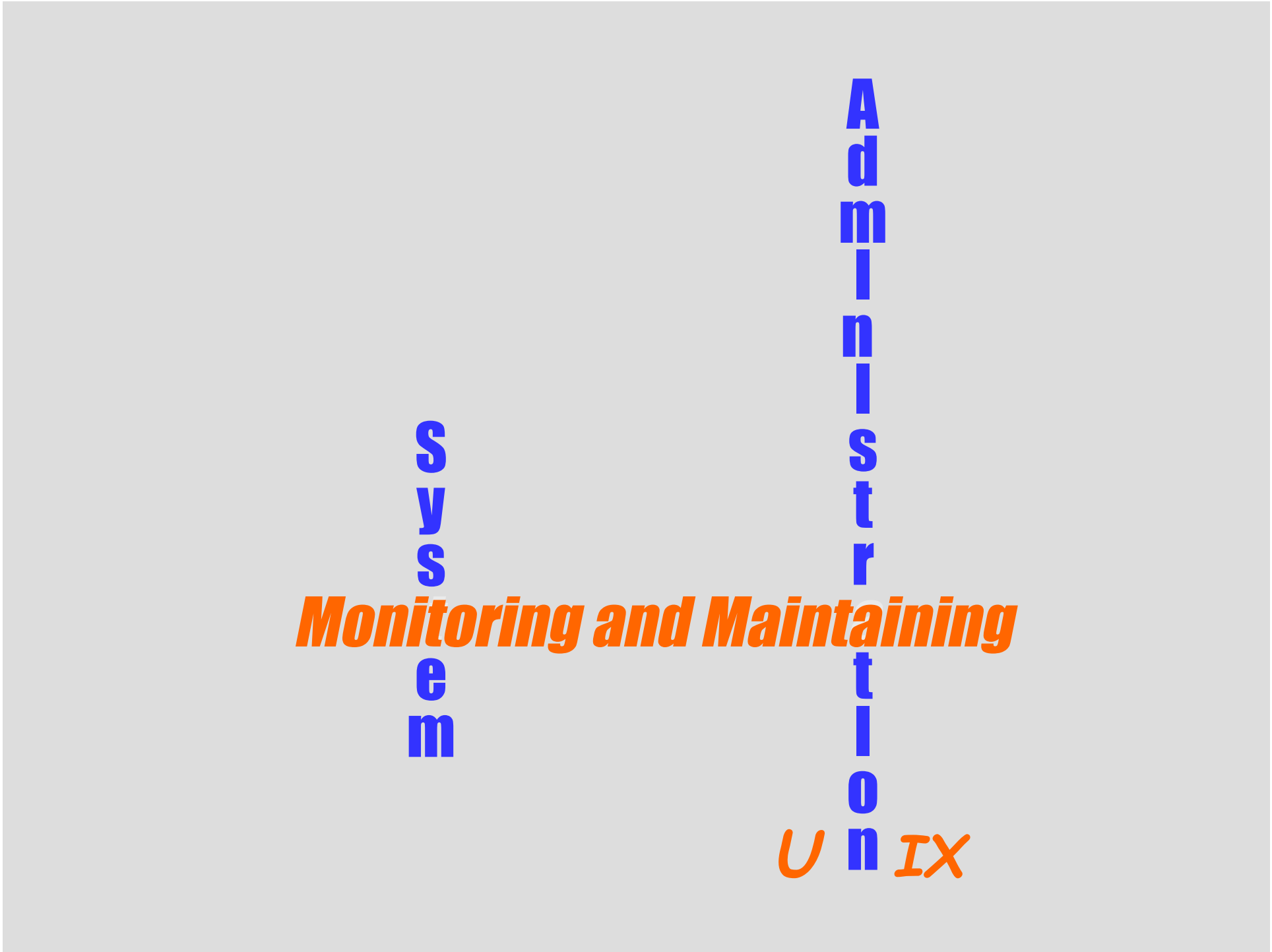

System Administration
Monitoring and Maintaining
UNIX

Syslog

- **support for system logging and kernel message trapping**
  - many modern programs use this facility to provide a standardized log facility
  - the kernel, device drivers and other core software also use syslog (klogd on Linux)
  - every logged message contains at least a time and a hostname field
    - *normally a program name field, too*
  - messages are structured according to:
    - *facility*
    - *importance*
  - quite configurable via /etc/syslog.conf
    - *once edited, notify syslogd via*
      - kill -HUP `cat /var/run/syslogd.pid`
  - *(see p.265, Frisch)*
  - logger application useful for shell scripts, etc.
    - (see p.266, Frisch)

Haiku Error Messages

⌘ http://www.salonmag.com/21st/chal/1998/02/10chal2.html

A crash reduces
your expensive computer
to a simple stone.

-- James Lopez

I'm sorry, there's -- um --
insufficient -- what's-it-called?
The term eludes me ...

-- Owen Mathews

To have no errors
Would be life without meaning
No struggle, no joy

-- Brian M. Porter

The code was willing,
It considered your request,
But the chips were weak.

-- Barry L. Brumitt

Serious error.
All shortcuts have disappeared.
Screen. Mind. Both are blank.

-- Ian Hughes

There is a chasm
of carbon and silicon
the software can't bridge

-- Rahul Sonnad

Logs And Distributed Logging

- **syslog logs to /var/log/messages**
  - listens to a socket (/dev/log) and the writes this file
  - (klogd listens to a 4k cyclic buffer in memory)

```
# tail /var/log/messages
Dec  8 20:29:57 redhat PAM_pwdb[339]: (login) session opened for user root by (uid=0)
Dec  8 20:29:57 redhat login[339]: ROOT LOGIN ON tty1
Dec  8 20:29:57 redhat PAM_pwdb[339]: (login) session closed for user root
Dec  8 22:10:06 redhat PAM_pwdb[420]: (su) session closed for user root
Dec  9 04:02:04 redhat PAM_pwdb[1039]: (su) session opened for user nobody by (uid=99)
Dec  9 04:03:53 redhat PAM_pwdb[1039]: (su) session closed for user nobody
Dec  9 06:32:46 redhat PAM_pwdb[1085]: (login) session opened for user bob by (uid=0)
Dec  9 06:32:46 redhat login[1085]: LOGIN ON ttyp0 BY bob FROM aunty
Dec  9 06:32:46 redhat PAM_pwdb[1085]: (login) session closed for user bob
Dec  9 06:45:48 redhat PAM_pwdb[1137]: (su) session opened for user root by bob(uid=0)
```

- **syslog can be configured to listen to messages sent over the network**
  - provides a centralized logging facility

```
# Sample syslogd configuration file to forward all
# messages to a remote host.
*.*           @hostname
```

  - use the -r switch to syslogd
  - to have this work correctly, /etc/services must contain the following entry:

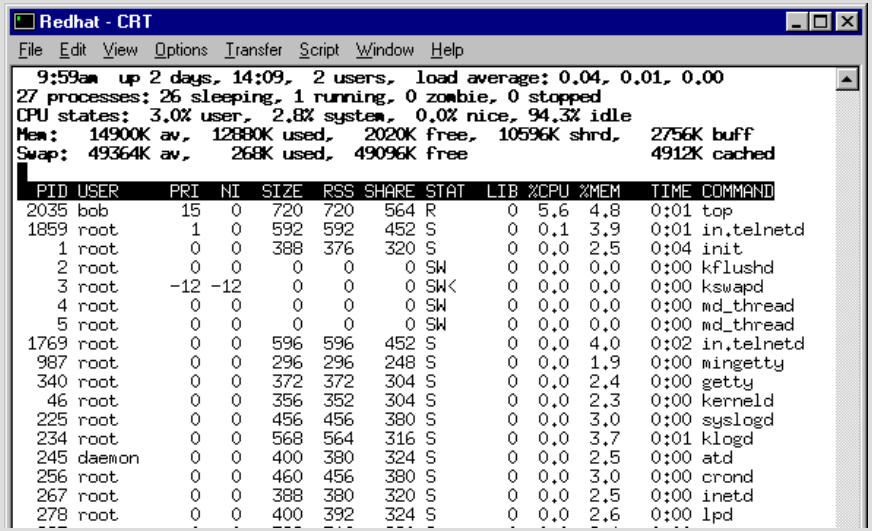
```
syslog          514/udp
```

Monitoring The System

- **Unix makes it easy to watch what is going on in the system…**
  - …but doesn't really provide the tools to tune things…
    - *the typical solution is to recompile the kernel*
    - *compare with a typical mainframe*
      - or (gasp!) Windows NT
- **monitoring is an intricate subject**
  - Schrödinger's cat…
- **what can be monitored**
  - CPU, disk space, memory (real and virtual)
- **some control mechanisms**
  - *(see p.275, Frisch)*

Basic Monitoring Tools

- CPU
  - uptime/w
  - ps
  - pstree
  - top
- disk
  - du/df
  - find
- most tools now examine /proc




```
# pstree
init-+-atd
     |-crond
     |-getty
     |-gpm
     |-httpd---2*[httpd]
     |-inetd-+-in.telnetd---tcsh---pstree
     |       `-in.telnetd---tcsh---man---sh-+-gunzip
     |                                      `-less
     |-kerneld
     |-kflushd
     |-klogd
     |-kswapd
     |-lpd
     |-2*[md_thread]
     |-2*[mingetty]
     |-nmbd
     |-smbd
     |-syslogd
     `-update
```

```
# cat /proc/meminfo
        total:    used:    free:  shared: buffers:  cached:
Mem:  15257600 12050432  3207168 10539008  1638400  5320704
Swap: 50548736   274432 50274304
MemTotal:      14900 kB
MemFree:        3132 kB
MemShared:     10292 kB
Buffers:        1600 kB
Cached:         5196 kB
SwapTotal:     49364 kB
SwapFree:      49096 kB
```

```
# w
  9:58am  up 2 days, 14:09,  2 users,  load average: 0.08, 0.02, 0.01
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU  WHAT
bob      ttyp0    aunty             8:44am  0.00s  3.42s  0.15s  w
bob      ttyp1    aunty             9:33am  2:15   1.81s  1.81s  -tcsh
```

```
# du -s /home/bob
32949   /home/bob
```

vmstat

- virtual machine statistics:
  - procs
    - *r: processes waiting for run time*
    - *b: processes in uninterruptable sleep*
    - *w: processes swapped out but otherwise runnable*
  - memory (kB)
    - *swpd: virtual memory used*
    - *free: idle memory*
    - *buff: memory used as buffers*
  - swap (kB/s)
    - *si: memory swapped in from disk*
    - *so: memory swapped to disk*
  - IO (blocks/s)
    - *bi: Blocks sent to a block device*
    - *bo: Blocks received from a block device*
  - system
    - *in: interrupts per second, including the clock*
    - *cs: The number of context switches per second*
  - CPU (percentages of total CPU time)
    - *us: user time*
    - *sy: system time*
    - *id: idle time*
  - *(see p.287, Frisch)*

```
# vmstat 5
 procs                  memory    swap        io     system         cpu
 r b w   swpd   free   buff cache   si   so   bi   bo   in   cs  us  sy  id
 0 0 0    268   1732   3012  4952    0    0    0    0  102    4   0   0 100
 0 0 0    268   1800   3012  4952    0    0    0    0  103   10   0   2  98
 0 0 0    268   1800   3012  4952    0    0    0    1  107    4   1   1  98
 0 0 0    268   1800   3012  4952    0    0    0    0  103    4   1   1  98
```

free

- free
  - a little simpler to understand than vmstat
    - *but only examines memory*

```
# free -s 5
             total       used       free     shared    buffers     cached
Mem:         14900      13068       1832      10316       3012       4944
-/+ buffers/cache:       5112       9788
Swap:        49364        268      49096

             total       used       free     shared    buffers     cached
Mem:         14900      13072       1828      10352       3012       4944
-/+ buffers/cache:       5116       9784
Swap:        49364        268      49096

             total       used       free     shared    buffers     cached
Mem:         14900      13164       1736      10684       3012       4952
-/+ buffers/cache:       5200       9700
Swap:        49364        268      49096
```

Control Tools

- limited and primitive
  - nice/renice
    - *a process's requested priority*
    - *lower gets more CPU attention*
    - *users can be 'nice' to other users and mark a process as less important by setting a high nice number*
    - *only the super user can set a low nice number to give priority to a process*

```
% nice +5 my_long_job
% renice 0 3486
```

  - swapon
    - *used to specify devices on which paging and swapping are to take place*
    - *usually executed during system boot*
  - kill and killall
  - kernel configuration
  - buy more and bigger...
    - *CPU, Disk, RAM, etc.*

Limits

- limit/ulimit
  - csh/bash builtin command
  - can be set by administrator
    - "Now for the bad news. Current UNIX resource limits are completely useless … for several reasons. First, the hard limits are often hard-wired into the kernel and cannot be changed by the system administrator. Second, users can always change their own soft limits. All an administrator can do is place the desired commands into users' .profile or .cshrc files and hope. Third, the limits are on a per-process basis. Unfortunately, many real jobs consist of may processes, not just one. … Finally, in many cases, limits are not even enforced; this is probably most often true of the ones you probably care about the most: CPU time and memory use."

```
% limit -h
cputime         unlimited
filesize        unlimited
datasize        unlimited
stacksize       8192 kbytes
coredumpsize    unlimited
memoryuse       unlimited
descriptors     256
memorylocked    unlimited
maxproc         256
openfiles       256
```

```
$ ulimit -a
core file size (blocks)  1000000
data seg size (kbytes)   unlimited
file size (blocks)       unlimited
max memory size (kbytes) unlimited
stack size (kbytes)      8192
cpu time (seconds)       unlimited
max user processes       256
pipe size (512 bytes)    8
open files               256
virtual memory (kbytes)  2105343
```

Installing Software

- **a very variable task**
  - some packages come as installable binaries
  - some come as source code
  - some work, some…
- **each platform has to have its own way of doing things, of course...**
  - e.g. AIX's SMIT, Solaris' admintool, etc.
    - *there is no guarantee that a vendor will use these 'standard' mechanisms*
  - we'll look at RedHat's rpm
    - *seems to be being adopted as a 'standard' in the linux world*
  - also look at stow
    - *seems like A Good Thing to me*

RPM

- **RedHat Package Manager**
  - manages the maintenance of software packages
  - a package is an archive of files, and package information, including name, version, and description.
  - ten basic modes of operation
    - *install, query, verify, check package signature, uninstall, build, rebuild database, fix permissions, set owners and groups and show rc file*
  - can perform upgrades without overwriting config files, etc.
  - can do automatic dependency following
    - *if package X requires package Y, ensure that Y is installed before installing X*
  - rpm package format allows for the inclusion of digital signatures
    - *ensure that a package comes from a trusted source and hasn't been tampered with*
  - can install across an ftp link from the internet
    - *if package source is given as an ftp URL*

RPM Examples

```
# rpm -qip which-1.0-8.i386.rpm
Name        : which                 Distribution : Manhattan
Version     : 1.0                        Vendor : Red Hat Software
Release     : 8                     Build Date : Tue Apr 28 02:59:13 1998
Install date                : (not installed)   Build Host    : porky.redhat.com
Group       : Utilities/File        Source RPM : which-1.0-8.src.rpm
Size        : 7227                     License : distributable
Packager    : Red Hat Software <bugs@redhat.com>
Summary     : Finds a program 'which' is in one of the directories on your PATH
Description :
Give it a program name, and it tells you if it is on your 'PATH'.

For example, 'which ls' would print '/bin/ls', because the ls program,
which is in one of the directories listed in your PATH environment
variable, is located in the /bin directory.
```

```
# rpm -qlp which-1.0-8.i386.rpm
/usr/bin/which
/usr/doc/which-1.0
/usr/doc/which-1.0/Makefile
/usr/doc/which-1.0/blah
/usr/doc/which-1.0/blah/Makefile
/usr/doc/which-1.0/which.c
/usr/man/man1/which.1
```

```
# rpm -ivh dump-0.3-13.i386.rpm
dump                        ##################################################
```

```
# rpm -qf /usr/bin/which
which-1.0-8
```

stow

- ***"Stow is a tool for managing the installation of multiple software packages in the same run-time directory tree. One historical difficulty of this task has been the need to administer, upgrade, install and remove files in independent packages without confusing them with other files sharing the same filesystem space."***
  - /usr/local/bin may have perl 5.004 files, python files 1.5 and gcc 2.7.2.3 files
    - *should be split into three separate subdirectories*
      - makes it easier to upgrade each in isolation
      - also makes it easier to retain python 1.4 *alongside* python 1.5
    - *but would mean that path management would become awkward*
  - stow puts a package into its own directory and then populates the appropriate standard directories with symlinks so that the system maintains the unstowed appearance
    - *does clever tree folding and splitting to minimise changes*
  - *(see http://www.gnu.org/software/stow/stow.html)*

rdist

- **maintains identical copies of files over multiple hosts**
  - very useful for updating system configuration files
    - *can be used to distribute updated programs (and anything else...)*
  - uses rsh to make connections to remote hosts
  - tasks are driven via a 'distfile'
    - *something like a 'makefile'*
    - *provides a rich set of configuration options*
      - update iff newer, iff binary comparison fails, etc.
      - send an email notification after doing something, log to syslog, etc.
      - maintain exception lists
      - do post-installation processing
      - etc.

rdist Example

```
# distfile
HOSTS = ( localhost )

FILES = ( /home/bob/distfile )

${FILES} -> ${HOSTS}
        install -ocompare /tmp/bob/distfile;

${FILES} :: /home/bob/distfile.tstamp
        notify bob@redhat ;
```

```
% rdist
/home/bob/stamp.bob: /home/bob/distfile: file is newer
/home/bob/stamp.bob: notify  ( bob@redhat )
localhost: updating host localhost
localhost: redhat: /tmp/bob/distfile: updated
localhost: updating of localhost finished
/home/bob/stamp.bob: updating of /home/bob/stamp.bob finished
```

```
% mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/bob": 2 messages 2 new
>N  1 rdist@redhat.skewst.  Sun Dec 20 11:39  15/490   "files updated after S"
&
Message 1:
From bob  Sun Dec 20 11:39:15 1998
Date: Sun, 20 Dec 1998 11:39:14 +1000
From: rdist@redhat.skewst.home.net.au (Remote distribution program)
To: bob@redhat.skewst.home.net.au
Subject: files updated after Sun Dec 20 11:38:23 1998


/home/bob/distfile.tstamp: /home/bob/distfile: file is newer

&
```

Maintaining man

- man is the standard online help system
- standard manual pages stored in source format in a directory hierarchy under /usr/man
  - file for section 1's ls is stored in /usr/man/man1/ls.1
  - may be other locations as well
    - *MANPATH environment variable*
    - */etc/man.config (see man.conf (5)) file contains*
      - information on how to construct the search path for man
      - full path names for various programs (nroff, eqn, etc.) used by man
      - a list with uncompressors for files with a given extension
    - *when a page is compiled for viewing, it may be cached in a 'mirror' catman directory*
      - /var/catman under Linux
      - so that next viewing is faster & easier on the system
      - if the catman hierarchy doesn't exist caching is disabled

```
# tree -d /usr/man
/usr/man
|-- man1
|-- man2
|-- man3
|-- man4
|-- man5
|-- man6
|-- man7
|-- man8
|-- man9
`-- mann

10 directories
```

- man -k/whatis/apropos keyword search facility supported by the whatis database
  - maintained by makewhatis command
  - run regularly by cron

System Administration UNIX

man Source Files

- driven by 'macros' embedded in the raw text

```
# cat /usr/man/man1/which.1
.TH WHICH 1 LOCAL
.SH NAME
which \- show full path of commands
.SH SYNOPSIS
.B which
progname ...
.SH DESCRIPTION
.I Which
takes a series of program names, and prints
out the full pathname of the program
that the shell would call to
execute it.
It does this by simulating the shells searching of the
.B $PATH
environment variable.
.SH "SEE ALSO"
The exec(2,3) family.
```

```
# man which
Formatting page, please wait...

WHICH(1)                                                    WHICH(1)

NAME
       which - show full path of commands

SYNOPSIS
       which progname ...

DESCRIPTION
       Which  takes a series of program names, and prints out the
       full pathname of the program that the shell would call  to
       execute it.  It does this by simulating the shells search-
       ing of the $PATH environment variable.

SEE ALSO
       The exec(2,3) family.

                                                LOCAL                1
```

- *no hyperlinking*
  - predates HTML by a LONG time!
- *displayed form created by 'compiling' the source with a macro library*
  - the 'man' macro package
    - (see table, p.391, Frisch)
  - essentially nroff -man *file*
- every administrator needs to know how to write manuals!