Structure of Directories*

Boris Veytsman

May 31, 2001

*This document contains lecture notes for informal Unix seminar for ITT AES employees (Reston, VA). No information in this document is either endorsed by or attributable to ITT. This document contains no ITT Privileged/Proprietary Information.

Unix Filesystem Again

Making files is easy under the UNIX operating system. Therefore, users tend to create numerous files using large amounts of file space. It has been said that the only standard thing about all UNIX systems is the message-of-the-day telling users to clean up their files. *System V.2 administrator's guide*

"Everything is either a process or a file"

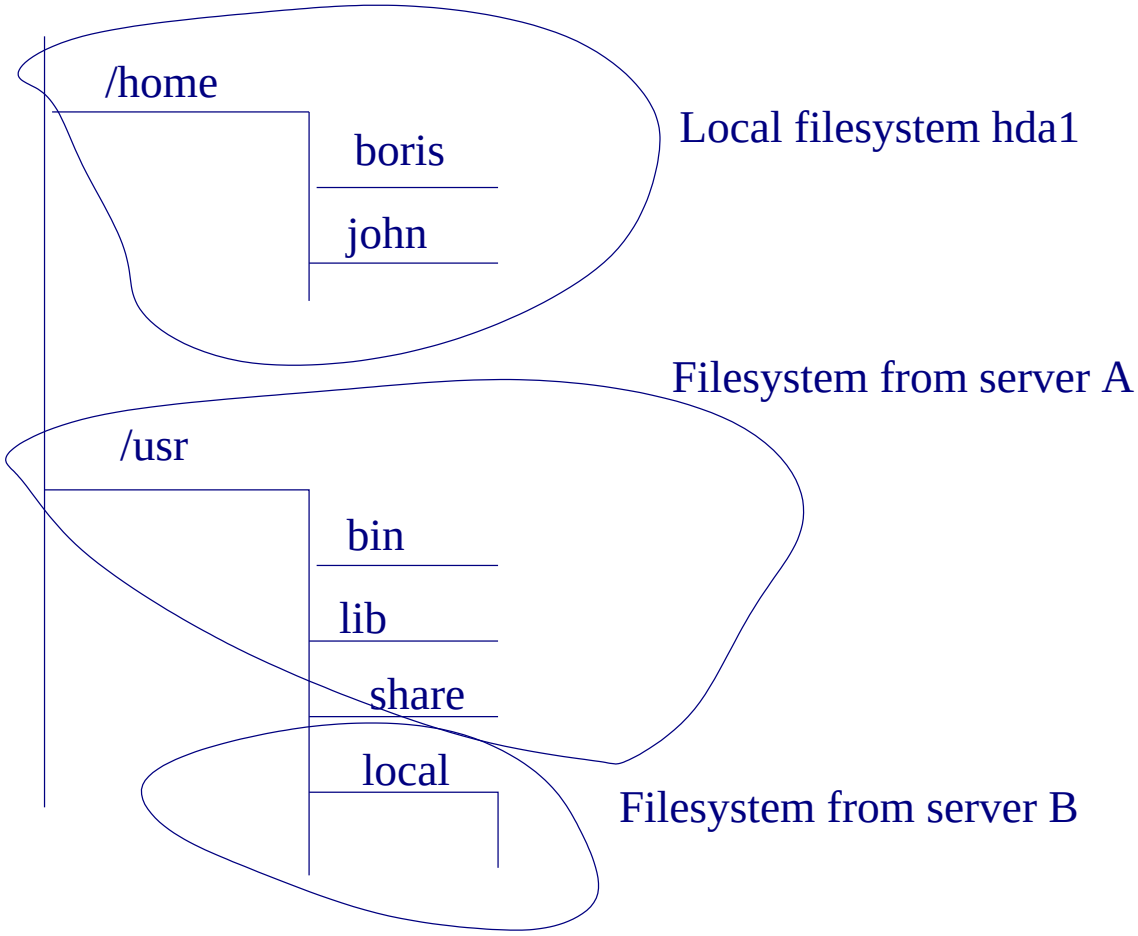
- "Traditional" files on media
- "Special files" for devices
- "Named pipes" for interprocess communications

Advantages:

1. Uniform API (`open`, `close`, `read` etc)
2. Uniform security model
3. Uniform hierarchical structure

Tree Structure

Theory is gray, but the golden tree of life is green. *Goethe*



ITT Industries
Engineered for life

- No drive letters, everything is in a tree
- The structure is determined by
  1. History
  2. Network considerations
  3. Backup considerations
- Absolute: `/home/boris/lib`. Relative: `lib/` or `../john/lib`

Root Filesystem

Money is the root of all money.

The only place where . equals . .

/bin

System binaries: `chown`, `cp`, `mv` etc.

Difference between `/bin` and `/usr/bin`: you can bring the system up and networking with only `/bin` mounted.

/sbin

Admin binaries: `fsck`, `shutdown`, `reboot` etc.

/lib

System libraries: `libc.so` etc.

/etc

- Configuration files
- Startup scripts
- In older systems—programs from /sbin

  Subdirectories: /etc/mail, /etc/init.d etc.

Admin notes:

1. Better to mount read-only
2. Backup when something changes
3. Have an extra floppy with `/etc`

/dev

Perilous to all of us are the devices of an art deeper than we ourselves possess. *J. R. R. Tolkien, "Lord of the Rings"*

A simple player:

```
cat yesterday.au > /dev/audio
```

A poor person's shredder:

```
yes > /dev/sda
```

A simple backup

```
tar -cvf /dev/tape ./
```

A bit bucket

```
verbose_program > /dev/null 2>&1
```

Mounting filesystems:

```
mount /dev/fd0 /floppy
```

/tmp

> There is nothing more permanent than a temporary building. There is nothing more permanent than a temporary tax. *Mix's Law*

Scratch files. Usual permission 1777 (sticky bit!)

Admin notes:

- Many programs write to it—must be large
- *Never* backup! Clean periodically.
- No quotas here for friendly environment
- Security concerns

/usr

> To the systems programmer, users and applications serve only to provide a test load. *Old Wisdom*

Place for user programs, libs, etc. Some dirs same as for /:

/usr/bin: For user binaries

/sbin: For admin progs

/lib: For libs

Why do not we use /usr/etc? Because of stupidity. . .

Some other interesting stuff:

`/usr/include`: C language includes

`/usr/share`: Architecture-independent files (for network shares): `man`, `lisp`, TEX. . .

`/usr/doc`: If you have it, do read it!

`/usr/games`: We do not have this, do we?[1]

[1] These lectures heavily use `/usr/games/fortune`

Admin notes:

1. Moderate backup schedule

2. Look for permissions here!

/usr/local

> A complex system that works is invariably found to have evolved from a simple system that works.

Locally built stuff. Once again same structure: `/usr/local/bin`, `/usr/local/sbin`, `/usr/local/lib`, `/usr/local/share`, `/usr/local/include`. But `/usr/local/etc` is often used!

Same as `/usr`, but for *local files*. Alternative location: `/opt`.

Admin notes:

1. Backup more often than `/usr`

2. Document everything!

Non-Standard Packages in `/usr` and `/usr/local`

> The nice thing about standards is that there are so many of them to choose from. *Andrew S. Tanenbaum*

Examples:

X11R6: `/usr/X11` (with `/usr/X11/bin`, `/usr/X11/include` etc)

Netscape: `/usr/local/netscape`

/var

One man's constant is another man's variable. *A. J. Perlis*

A place for variable system files:

`/var/log`: Log files

`/var/spool`: Mail and print jobs

`/var/run`: For PID files

etc

Admin notes:

1. Make it large!

2. Some pieces must be preserved (`/var/log`), others not (`/var/spool`)

/home

Making files is easy under the UNIX operating system. Therefore, users tend to create numerous files using large amounts of file space. It has been said that the only standard thing about all UNIX systems is the message-of-the-day telling users to clean up their files. *System V.2 administrator's guide*

Alternative locations: `/usr/home`, `/usr/people/`

Files owned by users:

- Personal stuff (`mail` etc.)
- Personal init files (`.cshrc`, `.profile` etc).

Shortcuts: `~` or `~boris`. Variable `$HOME`:

```
cd
cd ~
cd $HOME
cd ~john
```

Good practice: organize your home tree: `~/bin`, `~/lib`, `~/man` etc.

Add `~/bin` to `$PATH`, etc.

Admin notes:

1. Crossmount to all workstations
2. Use fast reliable filesystem on a *huge* disk
3. Backup daily. This is the most valuable part of the system.
4. A good place for group projects

Search Path and Other Variables

You can't go home again, unless you set `$HOME`

$PATH: where to search for programs.

```
export PATH=$HOME/bin:$PATH (sh)
setenv PATH $HOME/bin:$PATH (csh)
```

Dot in the current path: inherently insecure. Better *not* have it!

$MANPATH where man pages are

$PWD the currend directory

$HOME the home directory

```
boris@reston-0491:~/itt/unix/4_dirs$ cd $HOME
boris@reston-0491:~$ echo $PWD
/home/boris
```