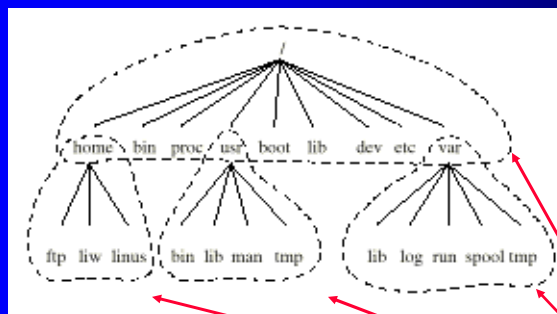


The landscape

David Morgan

File hierarchy overview

- A tree structure of directories
- The directory tree is standardized



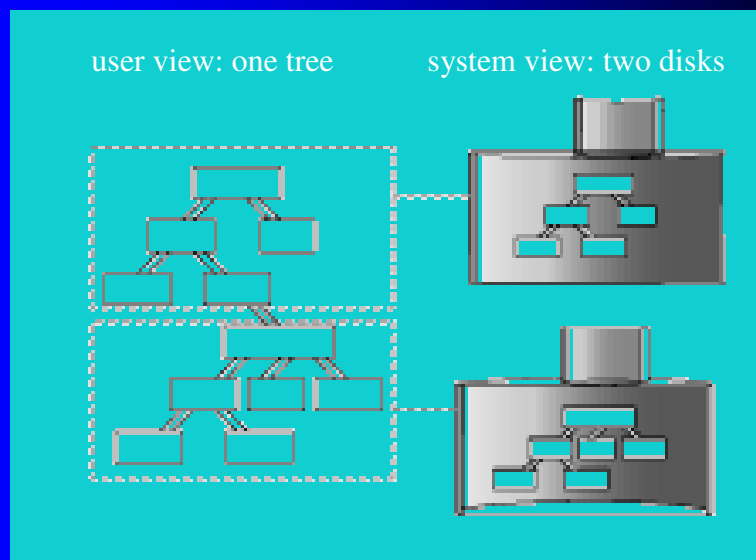
- But varies slightly among distributions

portions can spread
across
different partitions

Mounting file systems

- Grafting “subdirectories” onto file tree
- Integrates different devices’ subtrees into a single hierarchy
- Grafting location in tree called “mount point”
- Examples
 - transient mount of diskette drive to access floppy
 - mount a new hard drive to incorporate its capacity
- /etc/fstab for mount persistency

mounting



Toplevel directories

/bin	essential user command binaries (for use by all users)
/sbin	system binaries
/etc	host-specific system configuration
/root	home directory for the root user
/lib	essential shared libraries and kernel modules
/dev	device files
/tmp	temporary files
/boot	static files of the boot loader
/mnt	mount point for a temporarily mounted filesystem
/proc	pseudo filesystem exposing OS data structures
/usr	most installed programs
/var	dynamic system data
/home	users' home directories

A few important files

/etc/passwd	list of user accounts (without passwords!)
/etc/shadow	list of encrypted passwords for accounts
/etc/group	list of named groupings of accounts
/etc/fstab	what gets mounted where at boot
/etc/hosts	static table for host/address lookups
/etc/profile	script auto-run whenever a user logs in
/etc/resolv.conf	list of nameservers to use
/var/log/messages	“public” log file used by many apps

configs for individual server programs:

/etc/httpd/conf/httpd.conf	apache
/etc/samba/smb.conf	samba
/etc/ssh/sshdconfig	ssh

Permissions: ls -l shows them

```
[root@sputnik joe]# ls -l
total 52
drwxr-xr-x 4 joe joe 4096 Mar 13 06:05 Desktop/
-rw-r--r-- 1 david sales 524 Apr 8 21:30 adbudget
-rw-r--r-- 1 root root 30720 Apr 8 21:12 classifications.tar
-rw-r--r-- 1 joe sales 1620 Apr 8 21:29 mycalendar
-rw----- 1 joe joe 3302 Apr 8 21:29 mypaystubs
drwx----- 2 joe joe 4096 Mar 13 06:05 tmp/
[root@sputnik joe]#
```

permission set

ls -l shows a user & group associated with each file

```
[root@sputnik joe]# ls -l
total 52
drwxr-xr-x 4 joe joe 4096 Mar 13 06:05 Desktop/
-rw-r--r-- 1 david sales 524 Apr 8 21:30 adbudget
-rw-r--r-- 1 root root 30720 Apr 8 21:12 classifications.tar
-rw-r--r-- 1 joe sales 1620 Apr 8 21:29 mycalendar
-rw----- 1 joe joe 3302 Apr 8 21:29 mypaystubs
drwx----- 2 joe joe 4096 Mar 13 06:05 tmp/
[root@sputnik joe]#
```

user associated with file adbudget

group associated with file mycalendar

File system - permissions

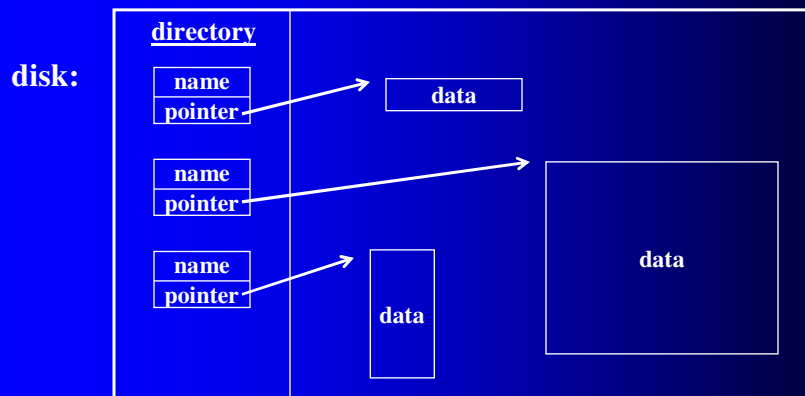
-rwxr-x---

- **File type** (file, directory, device,...)
- Accesses granted to **file's associated User**
- Accesses granted to members of **file's Group***
- Accesses granted to all **Other users**

*other than the associated user

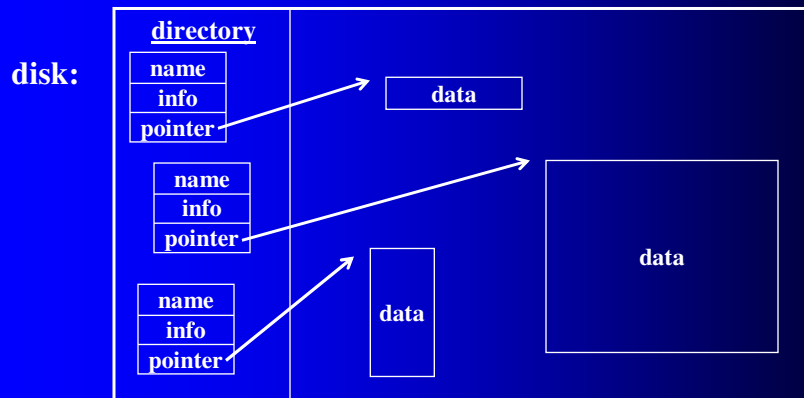
Logic of a filesystem structure

- purpose: associate names with bodies of data (aka "content")
- method: reserve part of the disk for a directory
- analogous to book's table of contents consuming first few pages

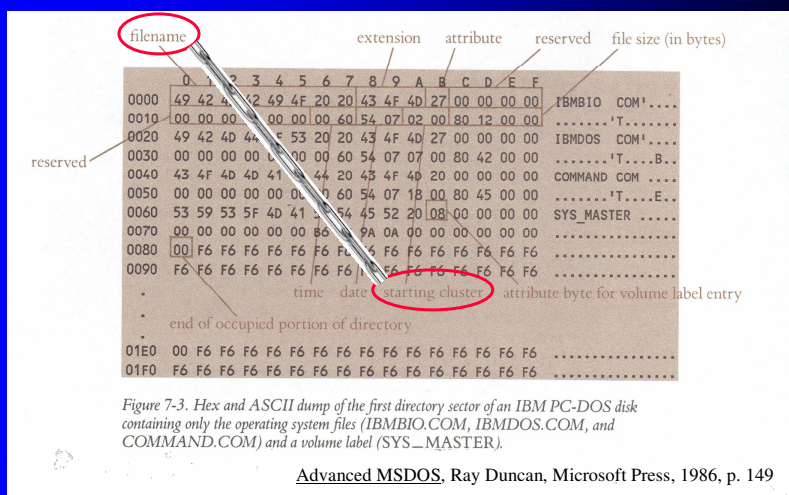


Logic of a filesystem structure

- directory entries may include characteristic file info
 - size, timestamp, filetype, etc.



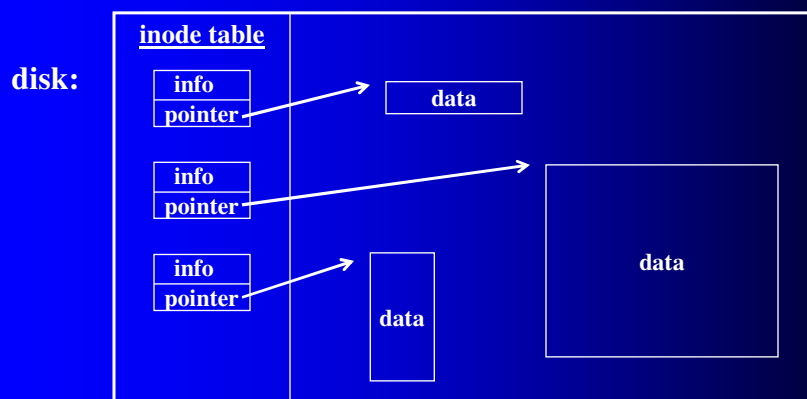
An example: FAT



Linux (ext2/3/4) filesystem structure

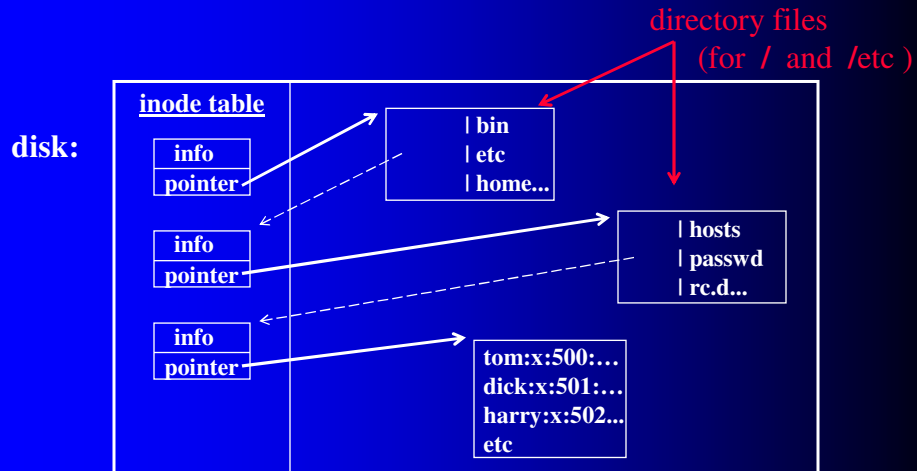
- directory portion called “inode table”
- table entries (inodes) lack files’ names!
- a “directory” is a regular file
- files’ names appear in directory files

Linux (ext2) filesystem structure -- but where are the filenames?



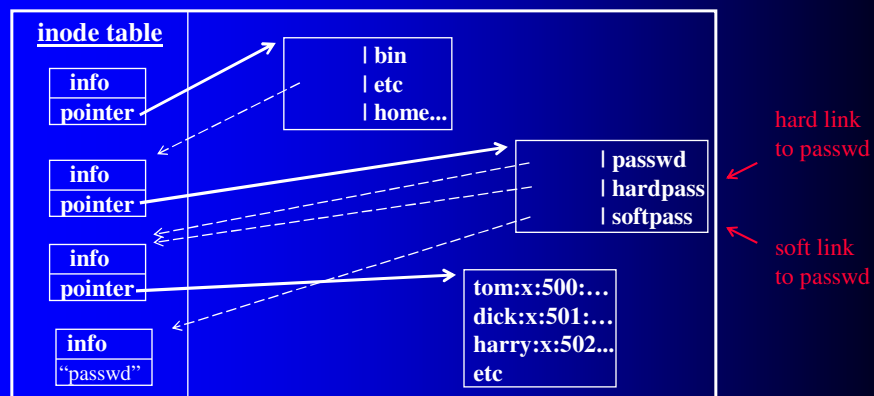
Directories, names

-- finding /etc/passwd



Hard and soft links

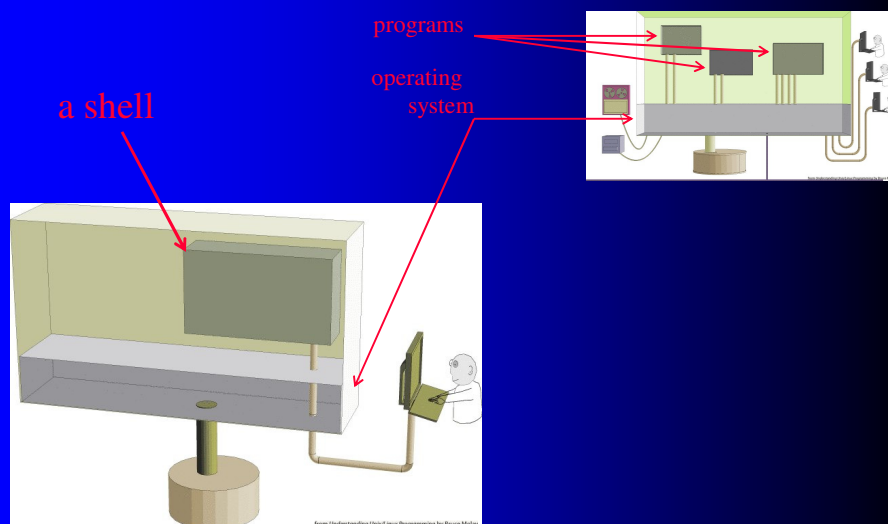
- hard: inode pointer values identical (`ln passwd hardpass`)
- soft: inode pointer to different inode, where filename occupies data pointer (`ln -s passwd softpass`)



What is “the shell?”

- A program
- /bin/sh
- or maybe /bin/bash or /bin/csh or /bin/zsh
 - or some other executable
 - determined for each user in /etc/passwd

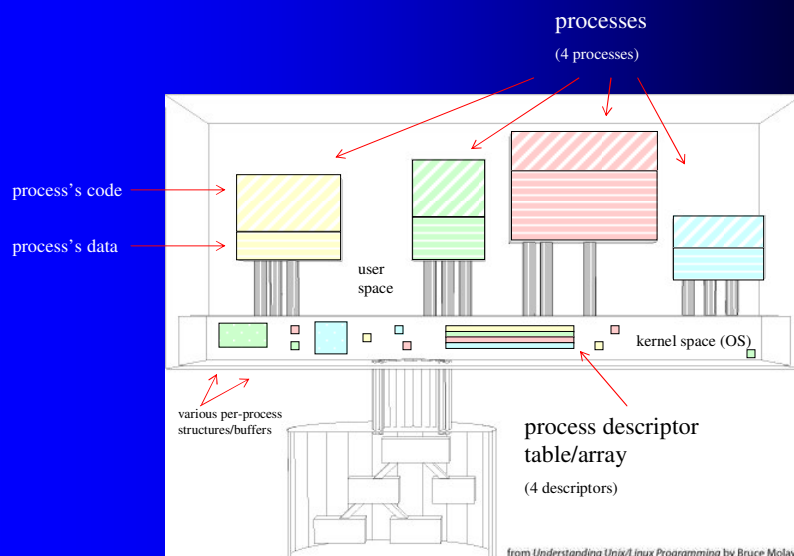
Not the operating system



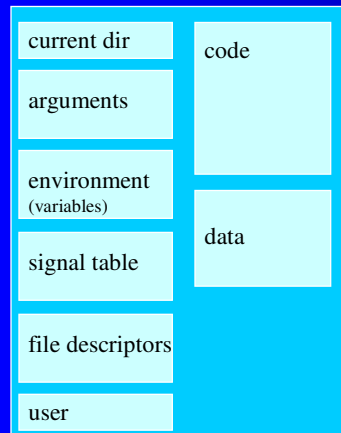
What does the shell do?

- Command processing
 - parse
 - expand
 - execute
- I/O redirection
- Piping
- Environment control
- Background processing
- Shell scripts

Process table tracks the processes



Single process in unix, consolidated view



Some important components

- code
- data
- current directory
- argument list
 - tokens from command line
- environment (variable) list
 - name=value pairs
- responses to signals
- list of open files
- user “as whom” process operates

ls -l foo bar

/home/david

0 = ls 1 = -l 2 = foo 3 = bar

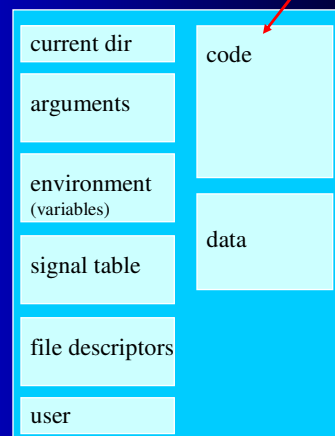
UID=500 LINES=25 PATH=...

1	default
2	handler pointer
3	ignore

0	stdin
1	stdout
2	stderr
3	/bin/ls

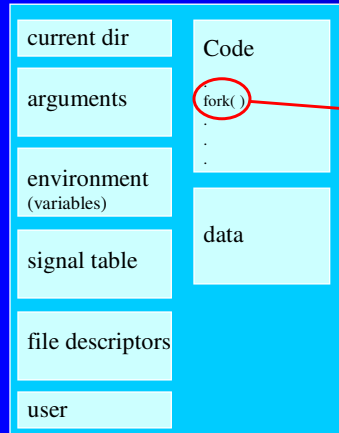
joe

process ID #1234

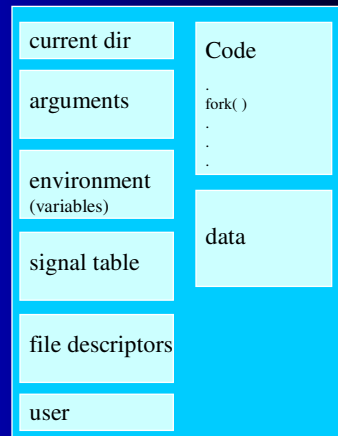


New process creation - fork()

process ID #1001



process ID #1002



same dir!
same args!
same vars!
same sigs!
same files!
same user!
even...
same code!!