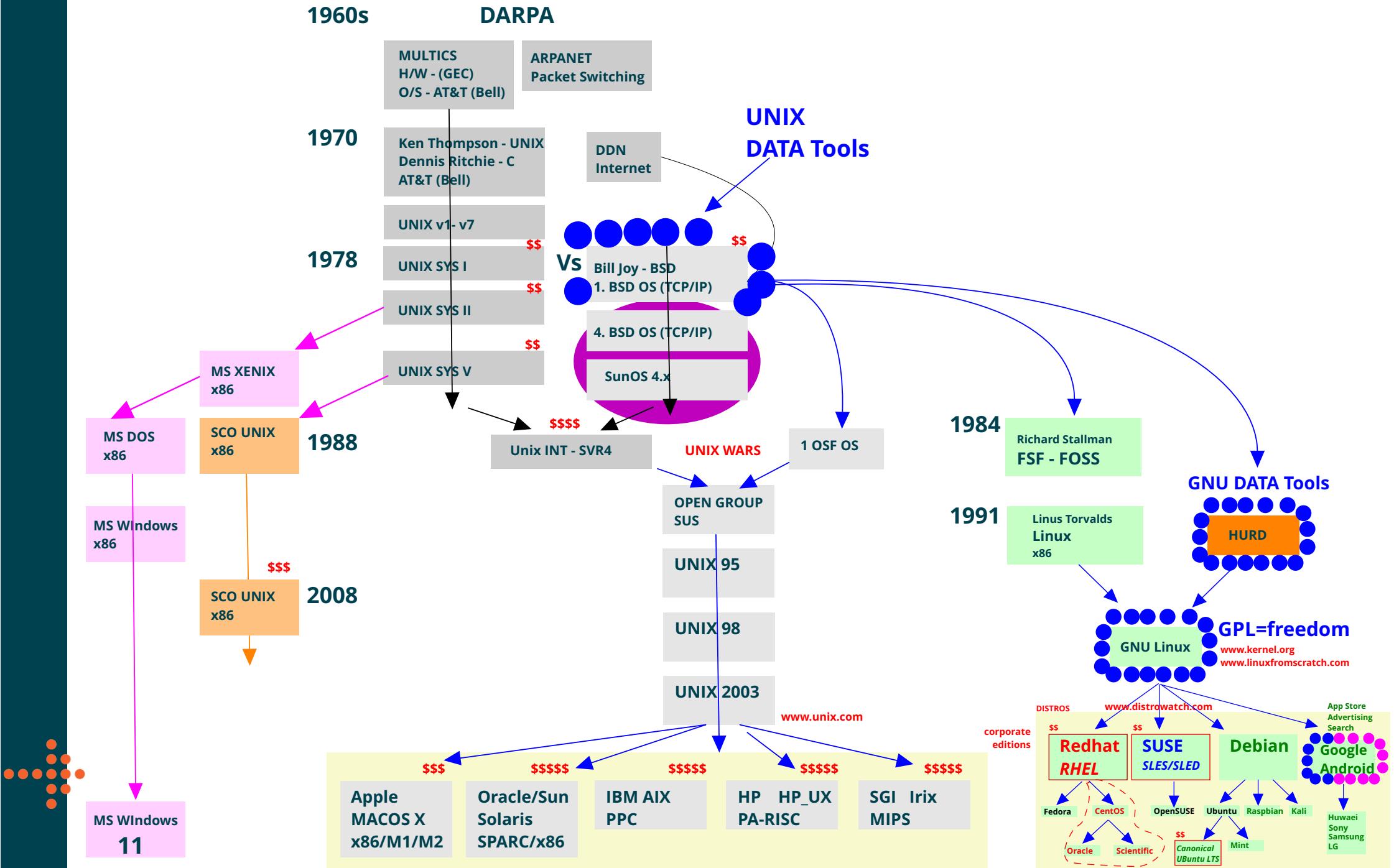
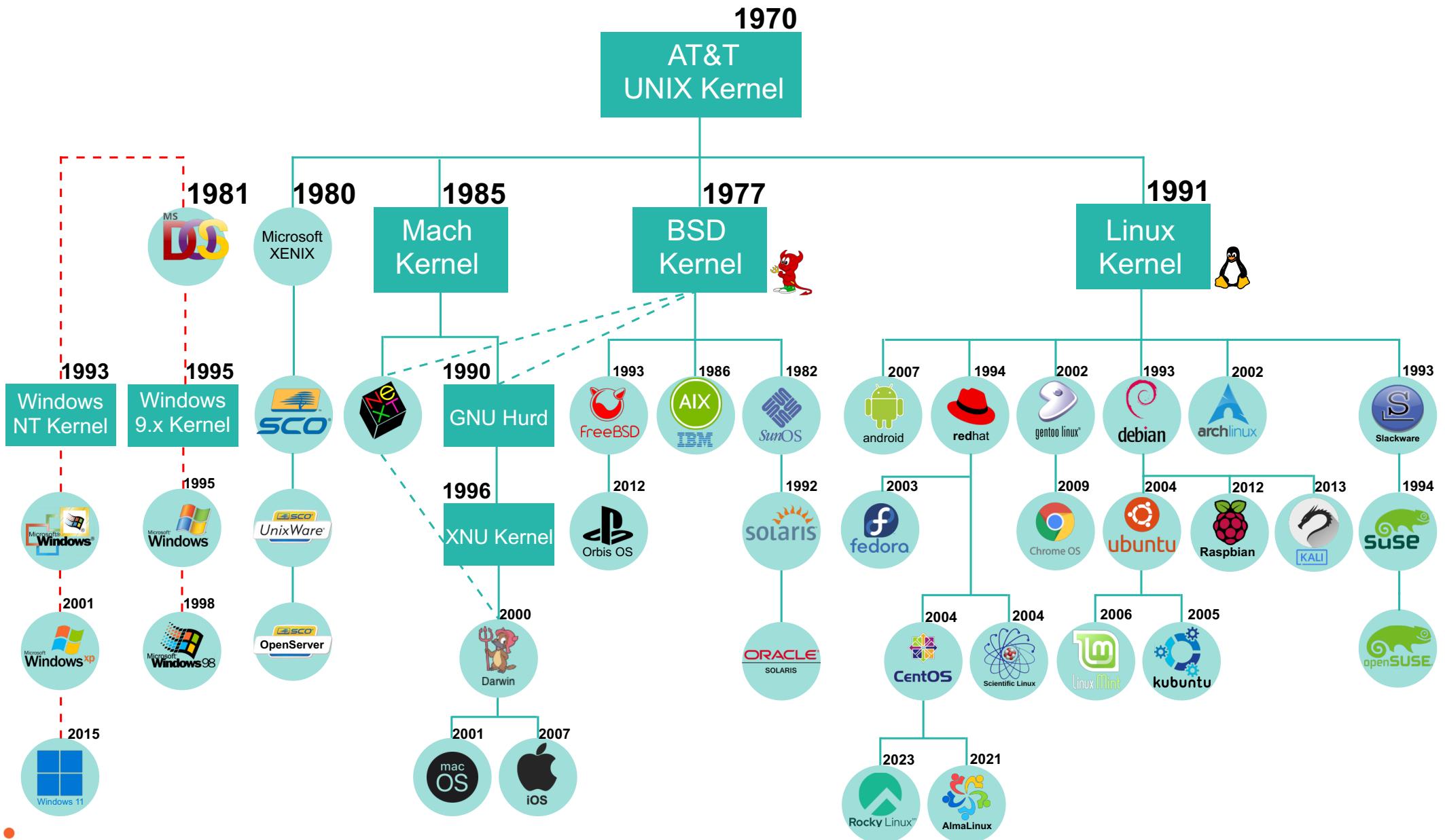


History of UNIX/Linux

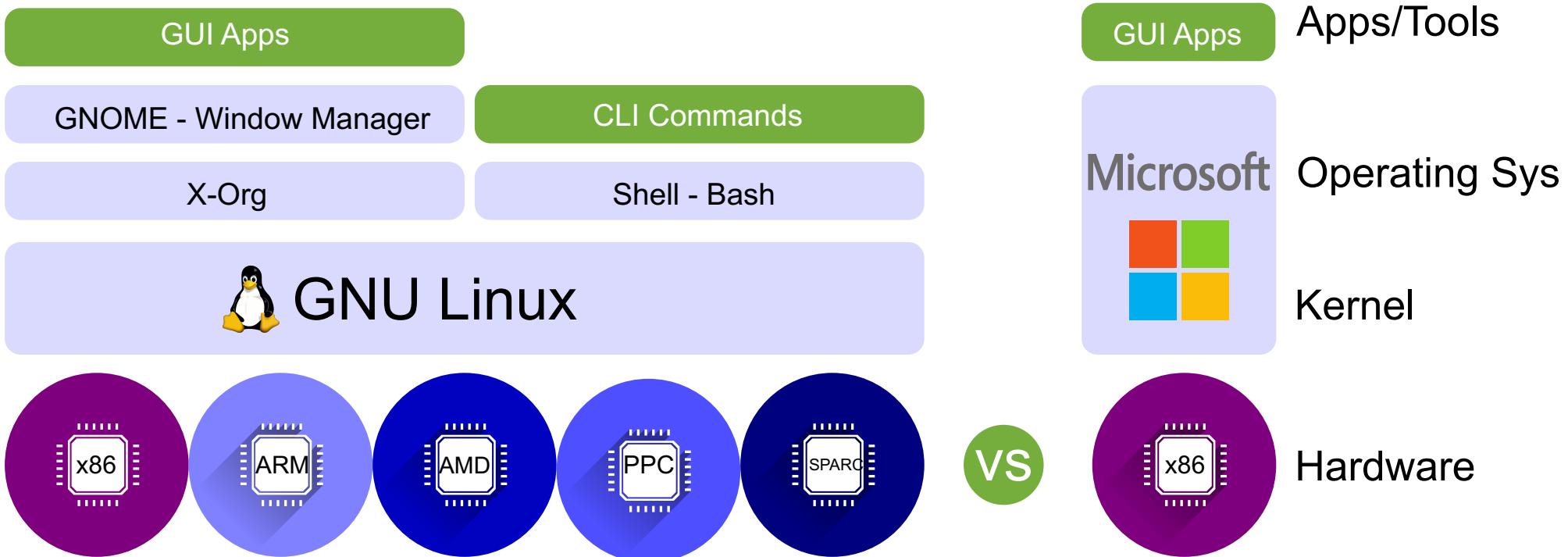


UNIX and Linux family tree



What is Linux?

A layer of software that sits between the software and the hardware.
The Kernel is the core of the “Operating System”



Why use Linux?



Who uses Linux?

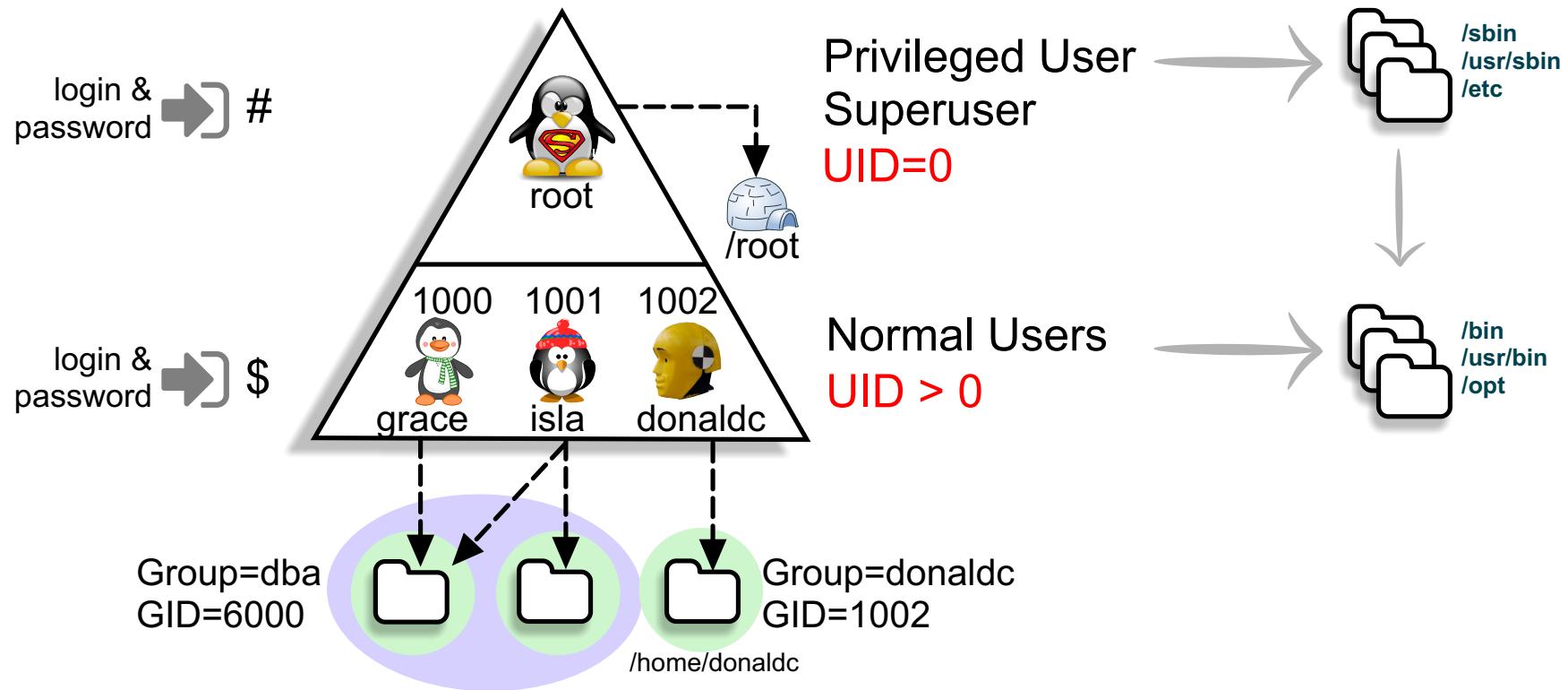


Internet
Finance
Insurance
Healthcare
Science
Government
Education
Military
Cyber
Gaming



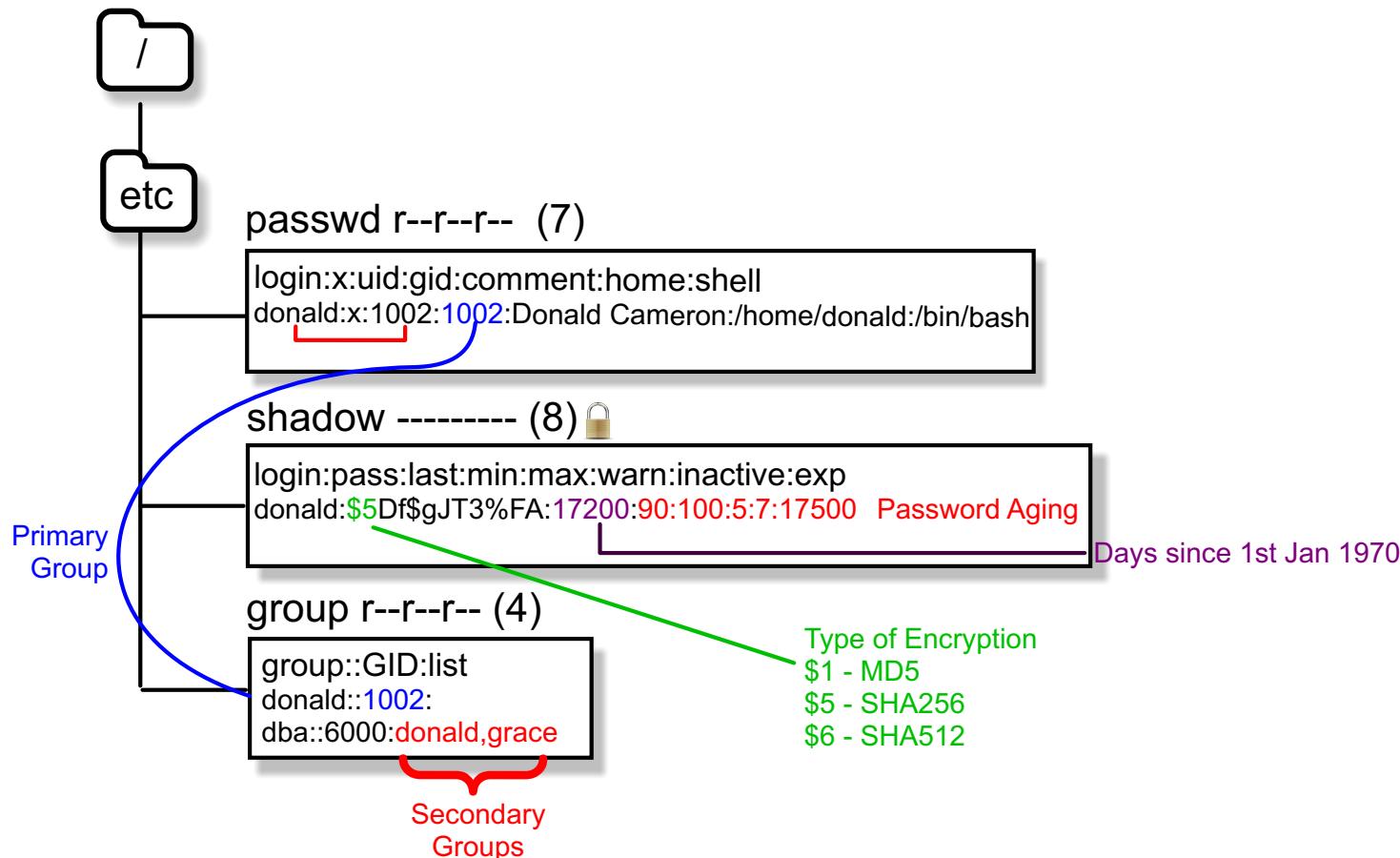
Linux is a CORE IT Skill!

UNIX/Linux User Accounts



Login: donaldc|dcameron|user1|student1
Password: min 7/8 chars, but preferably 14+
UID: [0 - 999] 1000 - 65k
GID: [0 - 999] 1000 - 65k {1x Primary Group
Home: /home/donaldc {128 x Secondary Group
Shell: /bin/sh
/bin/ksh
/bin/bash } x 1

UNIX/Linux User Accounts

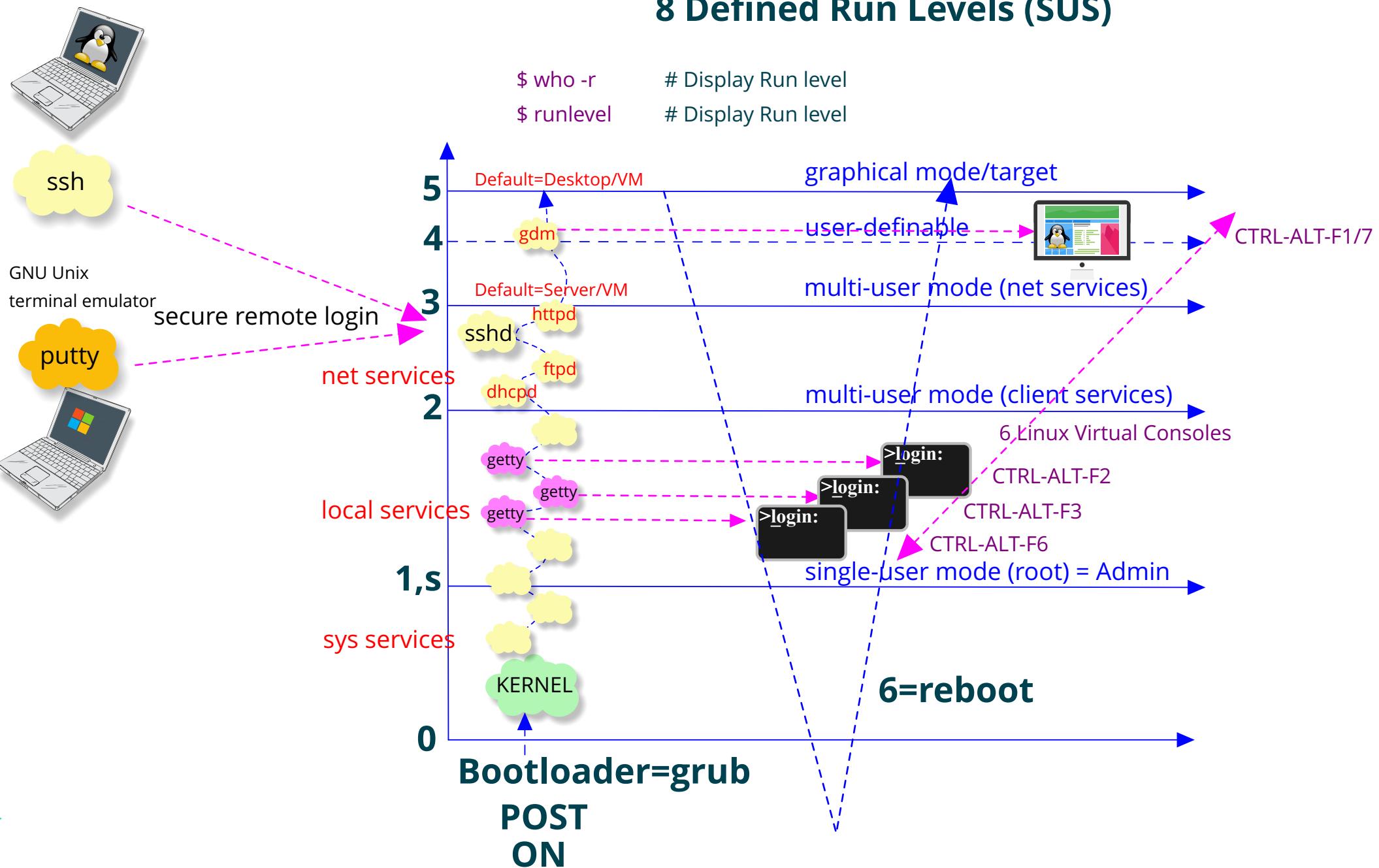


Further reading..
 \$ man 5 passwd
 \$ man 5 shadow
 \$ man 5 group

Further reading
 \$ man useradd|usermod|userdel
 \$ man groupadd|groupmod|groupdel
 \$ man passwd

How do you login?

8 Defined Run Levels (SUS)



Accessing the System

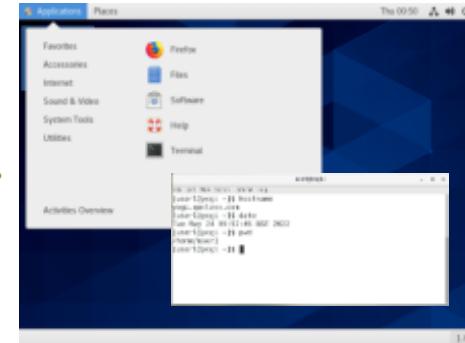
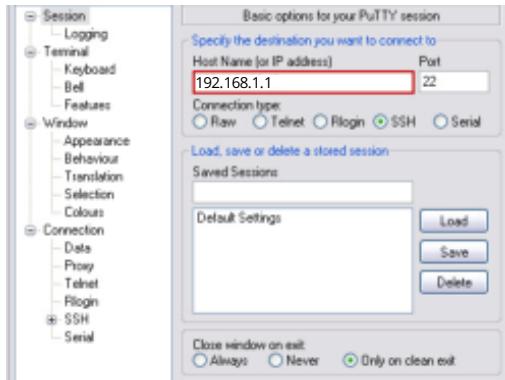
Gnome GUI Login



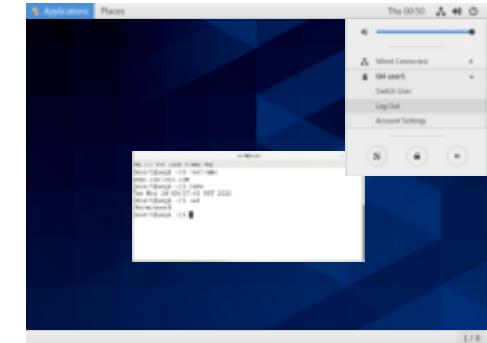
Local Console Login

```
Centos Core release 8.1  
yogi login: user1  
password:  
Last login: Mon May 1 16:43:20 on :0  
You have new mail.  
$
```

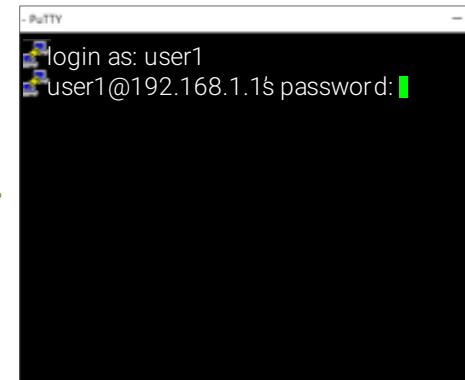
Putty SSH Remote Login



```
$ hostname  
yogi  
$ date  
Mon May 23 22:44:05 BST 2022  
$ pwd  
/home/user1  
$
```



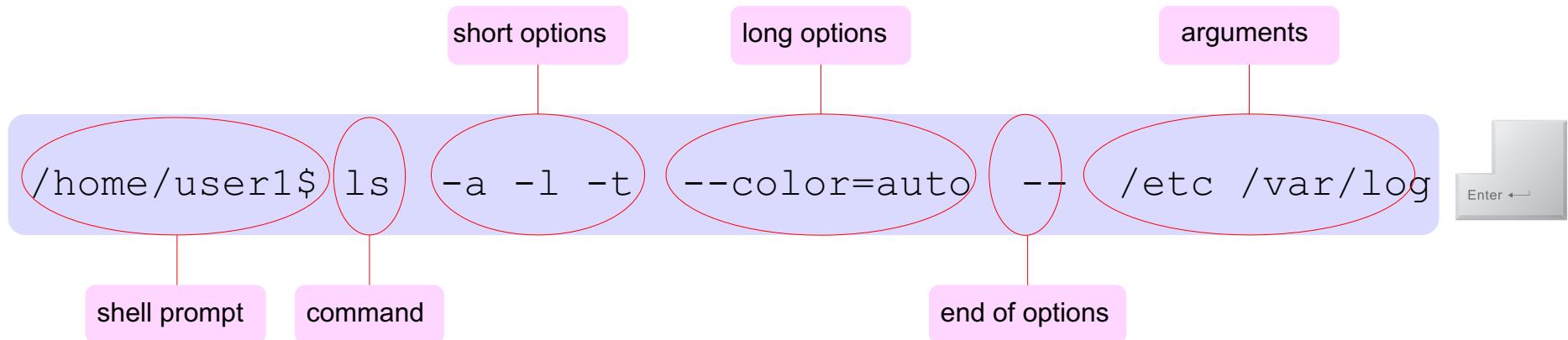
```
$ exit  
login:
```



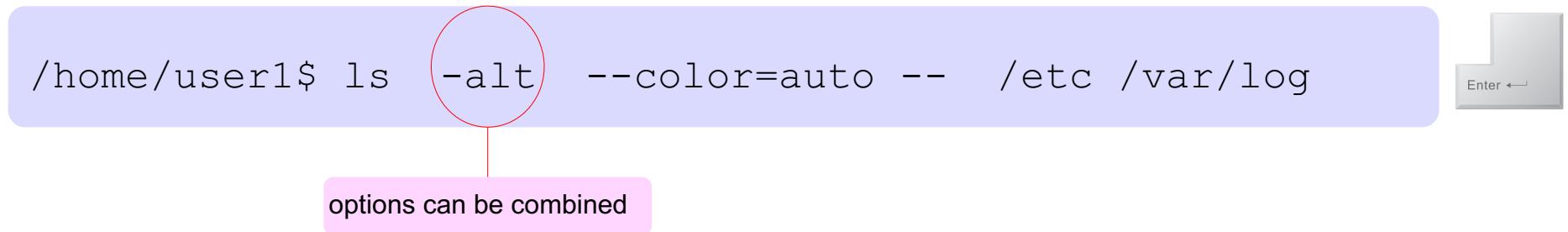
```
[user1@yogi ~]$ hostname  
yogi  
[user1@yogi ~]$ date  
Mon May 23 22:44:55 BST2022  
[user1@yogi ~]$ pwd  
/home/user1  
[user1@yogi ~]$ exit  
login:
```



Bash Shell - Entering commands



- Linux is CaSe sensitive - commands are in lowercase.
- Options can be single char (short) or descriptive name (long).
- Options end with '--' and arguments/parameters follow.
- Backspace or delete key to correct line.
- The long option `--help` displays list of options.



Bash Shell - Command line correction

```
/home/user1$ ls -alt --coxr=auto -- /etc /vxr/log
```

Diagram illustrating command line correction:

- Ctrl-U (^U)**: Deletes the word before the cursor.
- Ctrl-C (^C)**: Deletes the entire line.
- Ctrl-W (^W)**: Deletes the word after the cursor.



```
/home/user1$ stty -a
/home/user1$ stty erase ^?      # Enable Backspace
/home/user1$ stty erase ^h      # Enable Delete
```



A few commands to get you started/fact finding

\$ date	\$ hostname
\$ date +%j	\$ uname -n
\$ date +%d-%m-%y	\$ uname -a
\$ date +%d-%m-%Y	\$ uname -r
\$ date +%T	\$ uname -m
\$ date +%a	\$ id
\$ date +%A	\$ groups
\$ clear	\$ id root
\$ cal	\$ groups user10
\$ cal 7 1970	\$ grep 'user10' /etc/passwd
\$ cal 9 1752	\$ grep 'user10' /etc/group
\$ cal 2024	\$ who
\$ ls -a -l -t	\$ who -q
\$ ls -alt	\$ who -H
\$ ls -lta	\$ tty
\$ who -r	\$ exit
\$ runlevel	



Navigation and listing files and directories

```
$ cd /home/user1
```

```
$ cd $HOME
```

```
$ cd ~
```

```
$ cd
```

```
$ pwd
```

```
$ echo $PWD
```

```
$ ls -l
```

Mode	Links	user	user	size (bytes)	mtime	name
-rwxr-xr--	1	andrew	oracle	753	Fri 13 14:23	elm_street

Type
d = dir
- = file
l = Sym Link

```
$ ls -lt # Long listing in time order
```

```
$ ls -ltr # Long listing in reverse time order
```

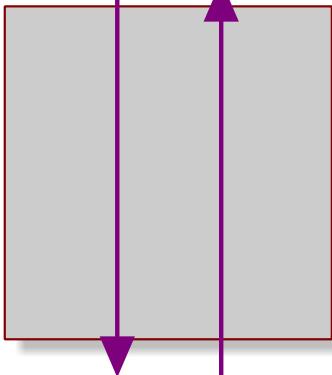
```
$ ls -altr # Long listing in reverse time order incl hidden files
```

```
$ ls -ld /etc # Long listing only of the parent dir
```

```
$ ls -lh # Long listing with size in human readable (kb(mb/gb))
```

View contents of text files

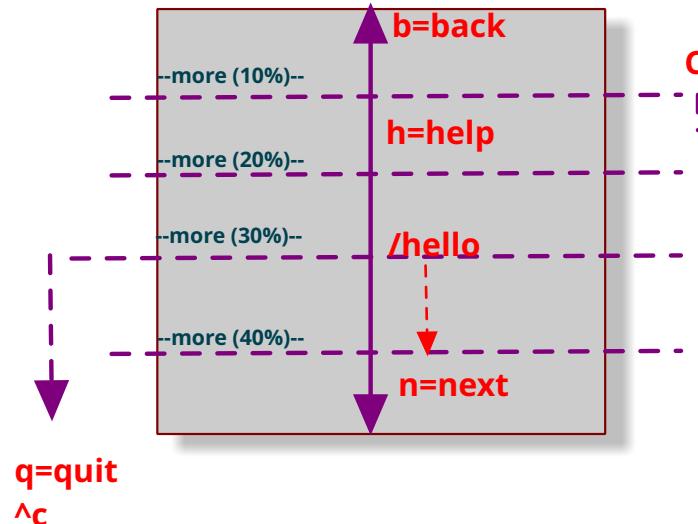
```
$ cat /usr/share/dict/words
$ cat -n romeo.txt
$ cat -b romeo.txt
```



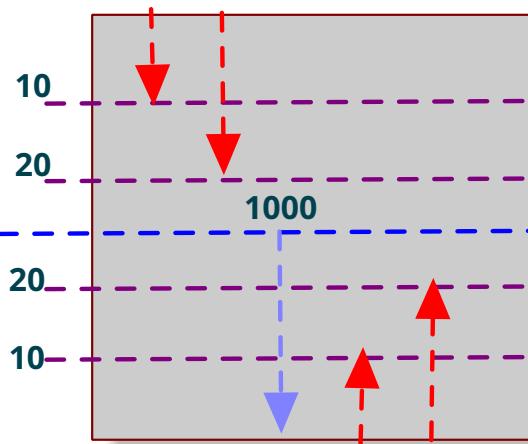
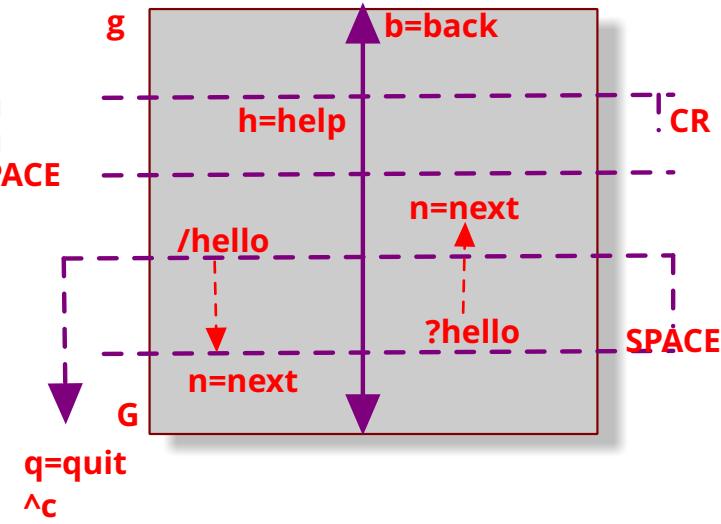
```
$ tac romeo.txt
$ wc /usr/share/dict/words
$ wc -l /usr/share/dict/words
$ wc -w /usr/share/dict/words
$ wc -c /usr/share/dict/words
```

```
$ tail -n +1000 /usr/share/dict/words
```

```
$ more /usr/share/dict/words
```



```
$ less /usr/share/dict/words
```



```
$ head /usr/share/dict/words
```

```
$ head -n20 /usr/share/dict/words
```

```
$ tail -n20 /usr/share/dict/words
```

```
$ tail /usr/share/dict/words
```

..and my favourite,

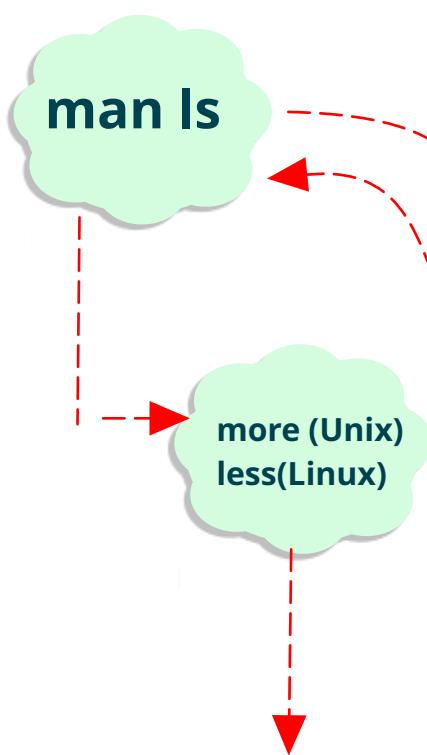
tail -f /var/log/secure

tail -f /var/log/auth.log

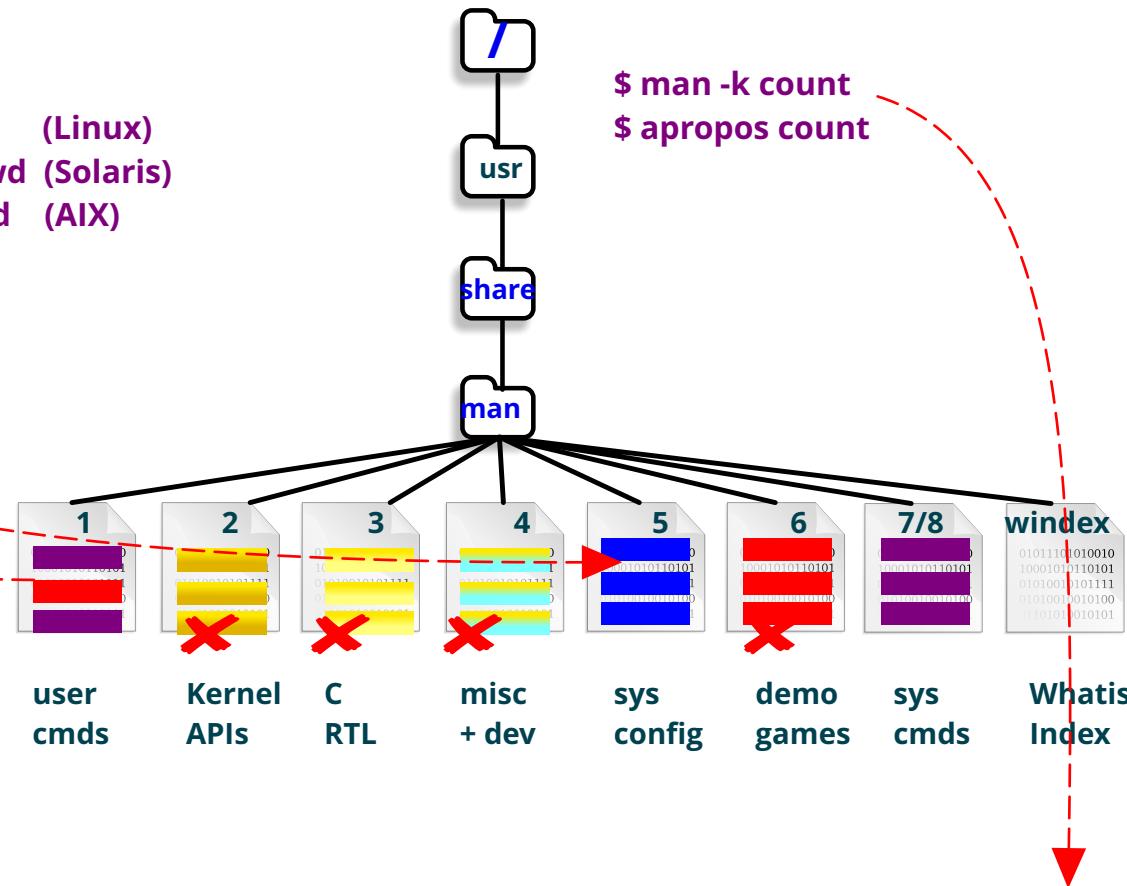
(^c to exit RHEL)

(Ubuntu/Debian)

Online Documentation/Programmers Ref Manual/man pages



\$ man passwd
\$ man 5 passwd (Linux)
\$ man -s5 passwd (Solaris)
\$ man -5 passwd (AIX)



return/space/b=back/h=help/q=quit

A screenshot of a terminal window displaying the man page for ls(1). The window shows the NAME, SYNOPSIS, DESCRIPTION, OPTIONS, and EXAMPLES sections of the documentation. A large green arrow points from the bottom left towards the terminal window.

```

LS(1)                               User Commands

NAME
ls - list directory contents

SYNOPSIS
ls [OPTION]... [FILE]...

DESCRIPTION
List information about the FILES (the current directory by default). Sort
is specified.

Mandatory arguments to long options are mandatory for short options too.

-a, --all
do not ignore entries starting with .

-A, --almost-all
do not list implied . and ..

--author
with -l, print the author of each file

-b, --escape
print C-style escapes for nongraphic characters

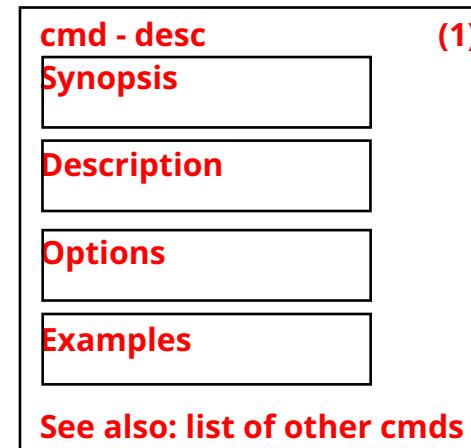
--block-size=SIZE
scale sizes by SIZE before printing them; e.g., '-block-size=M' prints
below

-B, --ignore-backups
do not list implied entries ending with .

-c
with -lt; sort by, and show ctime (time of last modification of f)

Manual page ls(1) line 1 (press h for help or q to quit)

```

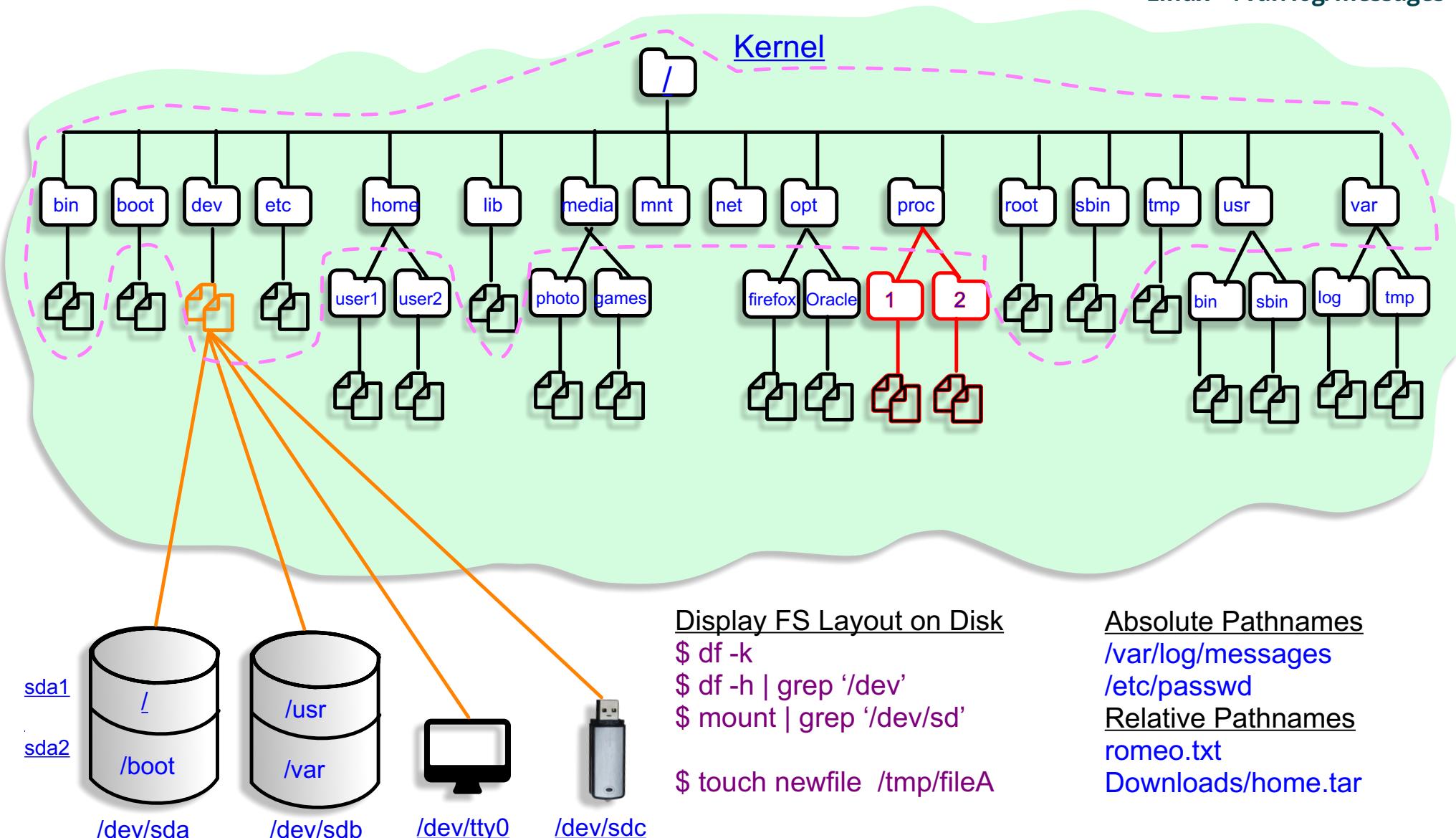


..and try \$ man man

catman -W

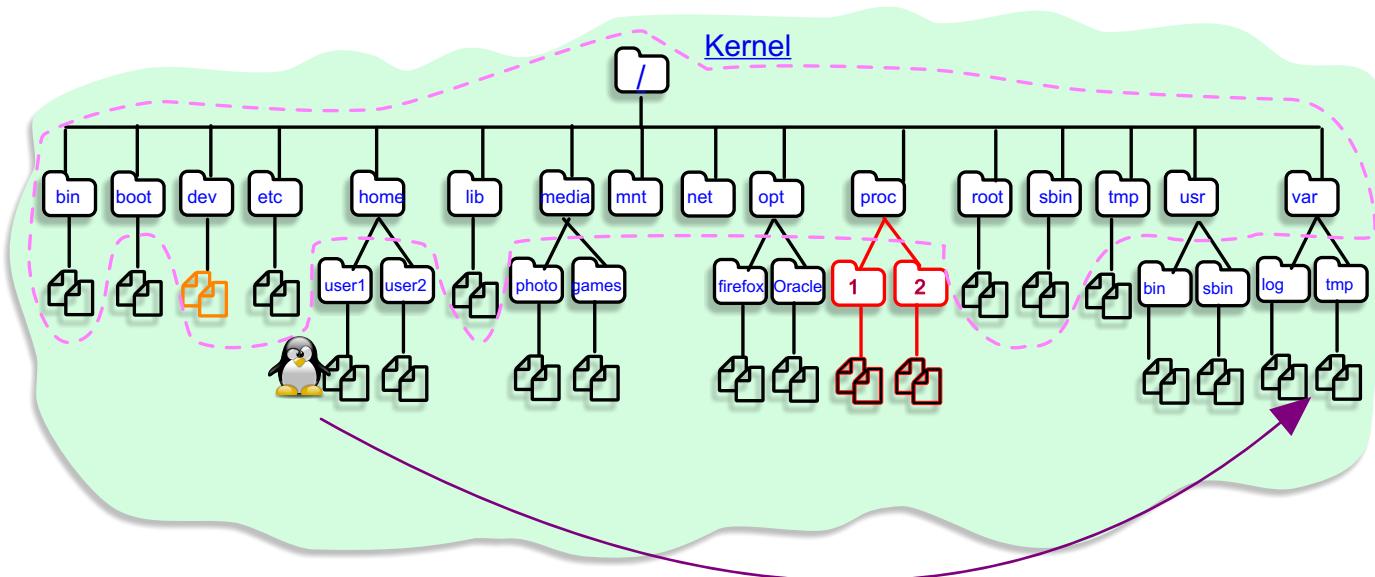
UNIX/Linux File System

Unix = /var/adm/messages
 Linux = /var/log/messages

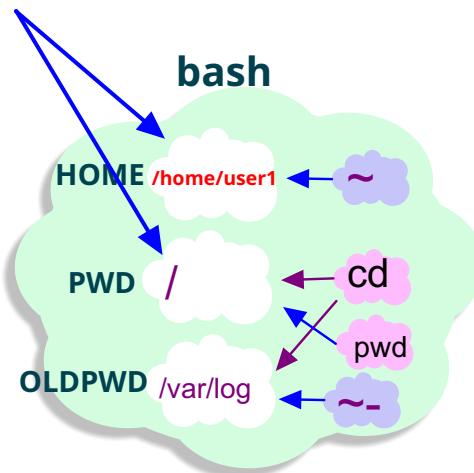


Source
<http://www.pathname.com/fhs>

Navigating the File System



/etc/passwd



Changing directory

```
$ cd /var/log
$ cd ../..
$ pwd
$ echo $PWD
$ cd /home/user1
$ cd $HOME
$ cd ~
$ cd
$ cd ~- # OLDPWD
```

Listing the contents of a directory

Type	Perm	Hard	User	Group	Size	Mtime	Name
-rwxr-xr--.	1	user10	QALXESS	453	Jul 14 10:52	romeo.txt	

\$ ls
\$ ls .
\$ ls --color=auto
\$ ls -F /etc
\$ ls -l

\$ ls -lh
\$ ls -lt

Long listing in human readable size.
Long listing in time order.

\$ ls -altr # Long listing of ALL in reverse time order
\$ ls -ltr /etc | more
\$ ls -R / # Long listing Recursively from '/'
\$ ls -ld ~



Managing your files & Dirs

Creating/Removing Dirs

```
$ mkdir memos ..reports /var/tmp/photos
```

```
$ mkdir a b c
```

```
$ mkdir a a/b a/b/c
```

```
$ mkdir -p a/b/c
```

```
$ rmdir memos ..reports /var/tmp/photos
```

Only removes empty directories!

```
$ rmdir -p a/b/c
```

```
$ rmdir -r ..reports
```

Warning removes directory and contents!



Copying files & dirs

```
$ cp romeo.txt juliet.txt
```

```
$ cp /etc/motd ./message
```

```
$ cp /etc/motd ~/message
```

```
$ cp /etc/motd ~/motd
```

```
$ cp /etc/motd ~
```

```
$ cp romeo.txt juliet.txt Documents
```

```
$ cp f1 f2 f3 f4 f5 dirX
```

\$ cp romeo.txt BigProdData

Careful will overwite existing files!



\$ cp -i romeo.txt BigProdData

overwrite (y/n)?



\$ cp -f romeo.txt BigProdData

Careful will overwite existing READONLY files!



\$ cp -r projectx newproject

Copy Directory and its contents!



Removing Files & Dirs

```
$ rm fileA ..fileB /var/tmp/fileC
```

```
$ rm *
```



Warning removes all files!

```
$ rm -i *.pdf
```

remove (y/n)?



\$ rm -f romeo.txt BigProdData

Warning removes READONLY files!



\$ rm -r projectx

Warning removes dir and contents!

rm -rf *



Moving/RENAMING files & dirs

```
$ mv romeo.txt juliet.txt
```

```
$ mv /etc/motd ./message
```

```
$ mv /etc/motd ~/message
```

```
$ mv /etc/motd ~/motd
```

```
$ mv /etc/motd ~
```

```
$ mv romeo.txt juliet.txt Documents
```

```
$ mv f1 f2 f3 f4 f5 dirX
```

\$ mv romeo.txt BigProdData

Careful will overwite existing files!



\$ mv -i romeo.txt BigProdData

overwrite (y/n)?



\$ mv -f romeo.txt BigProdData

Careful will overwite existing READONLY files!



\$ mv projectx newproject

Rename top Directory name!



Hard & Symbolic Links

kernel

bin	45
etc	715
dev	89
home	1024

/etc

group	800
motd	802
passwd	875
shadow	1200

/home/sean

kylie	100
runrig	110
proclaimers	500

/home/wayne

runrig	150
myKylie	100
message	200

bash

PWD

/home/sean

ls -l kylie
cat kylie

Hard Link

\$ cp ~sean/runrig ~/runrig



\$ ln ~sean/kylie ~/myKylie

Cannot Hard Link Directories!

Cannot Hard Link across FS partitions!

Symbolic Link \$ ln -s /etc/motd ~/message

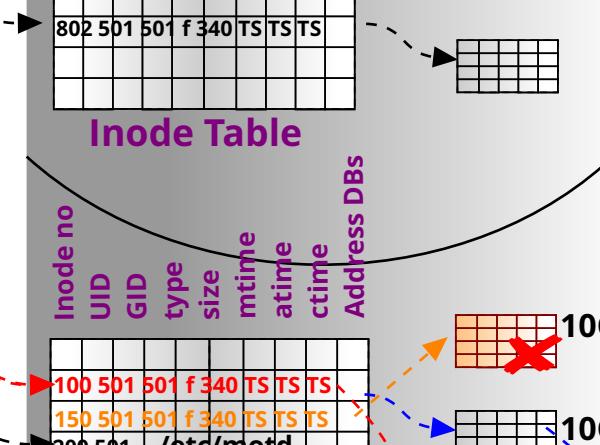
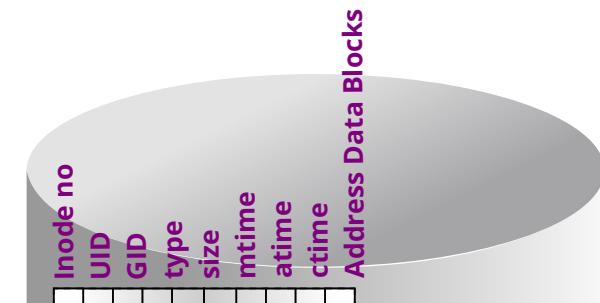
Can Link to directories!

Can link files across partitions!

Fact Finding

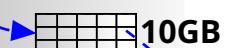
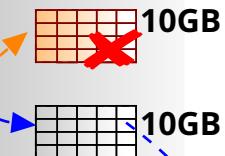
\$ ls -il /etc/motd

\$ stat /etc/motd



/

/home



vi/vim = 3 x Modes

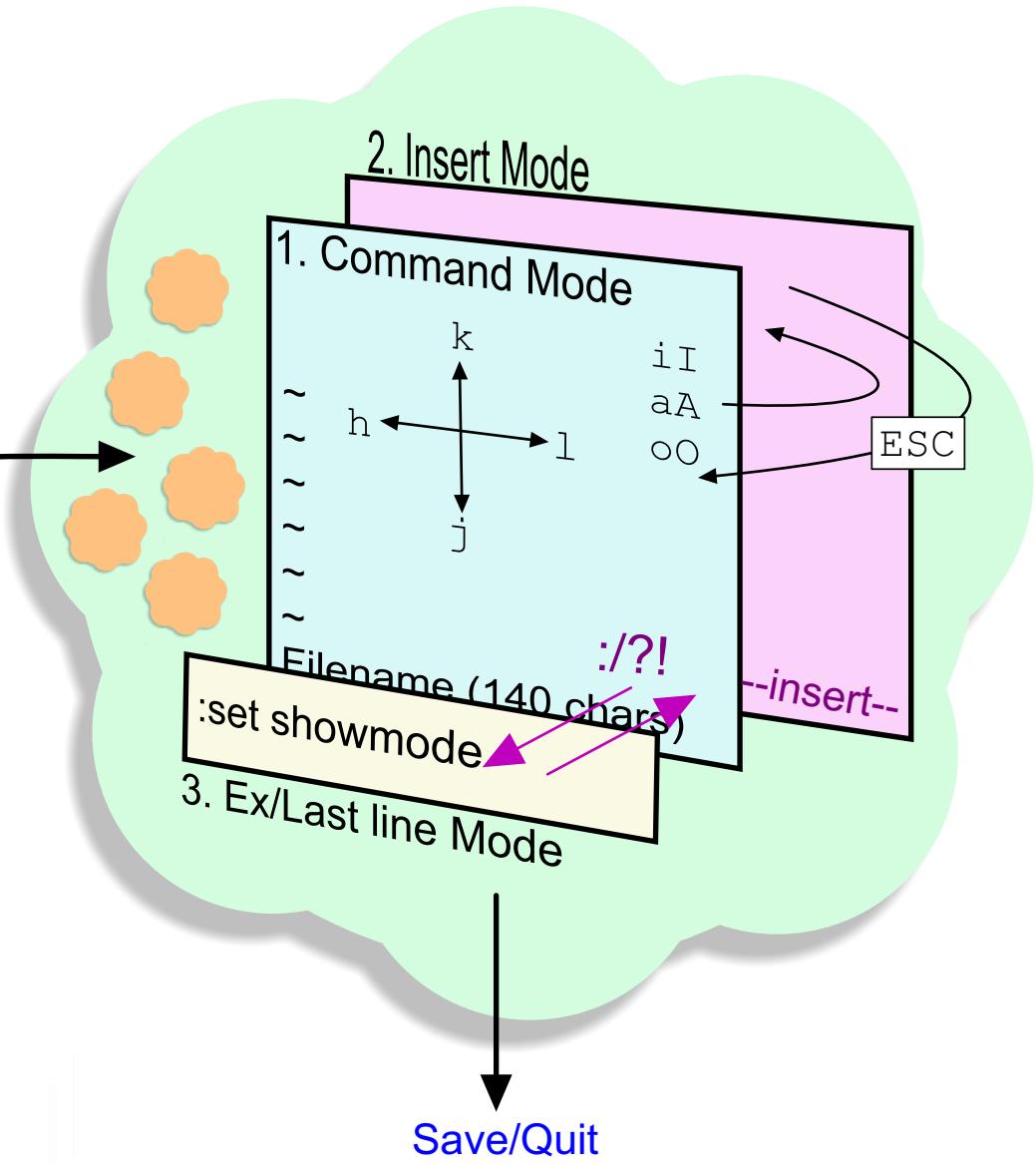
Starting vi

```
$ vi datafile
$ vi newfile
$ vi
$ view datafile
$ vi -r # Recover last file you were editing
```

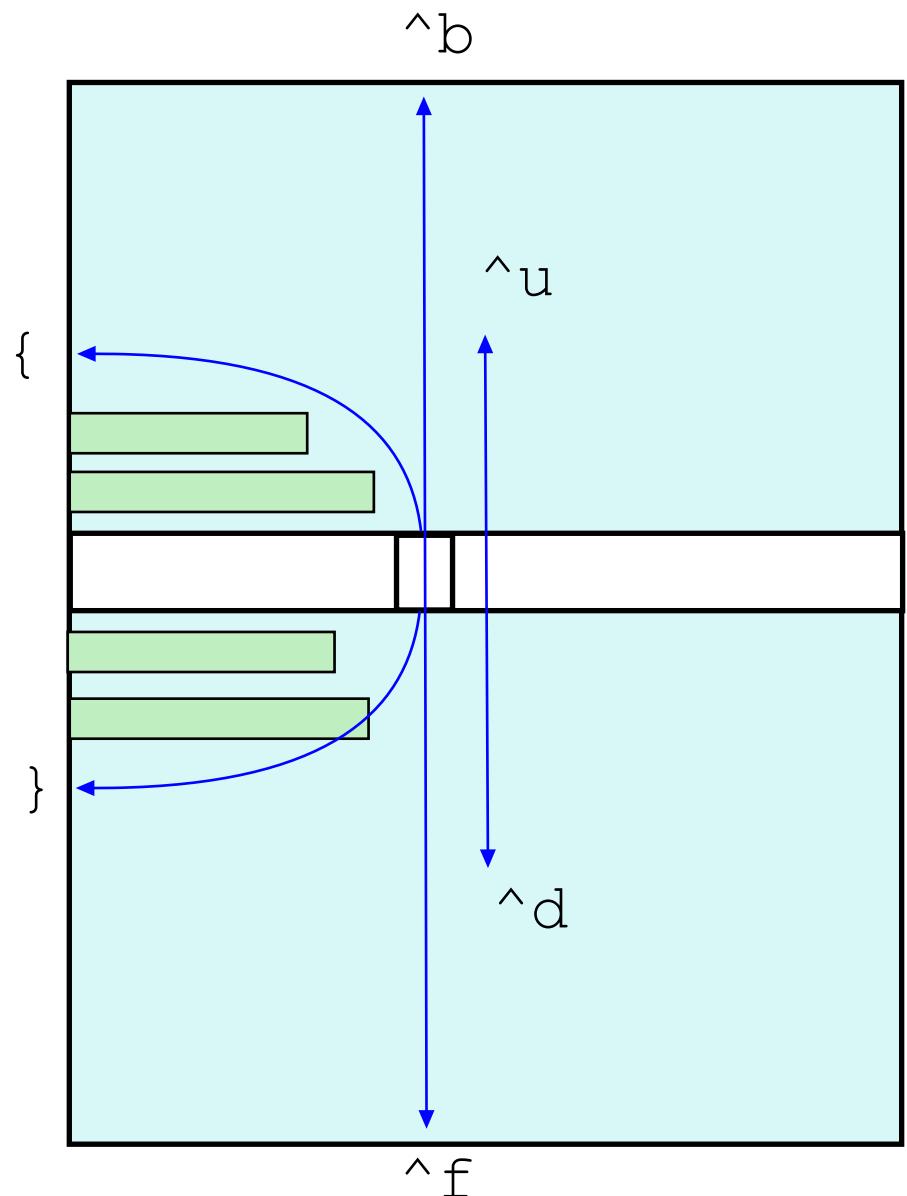
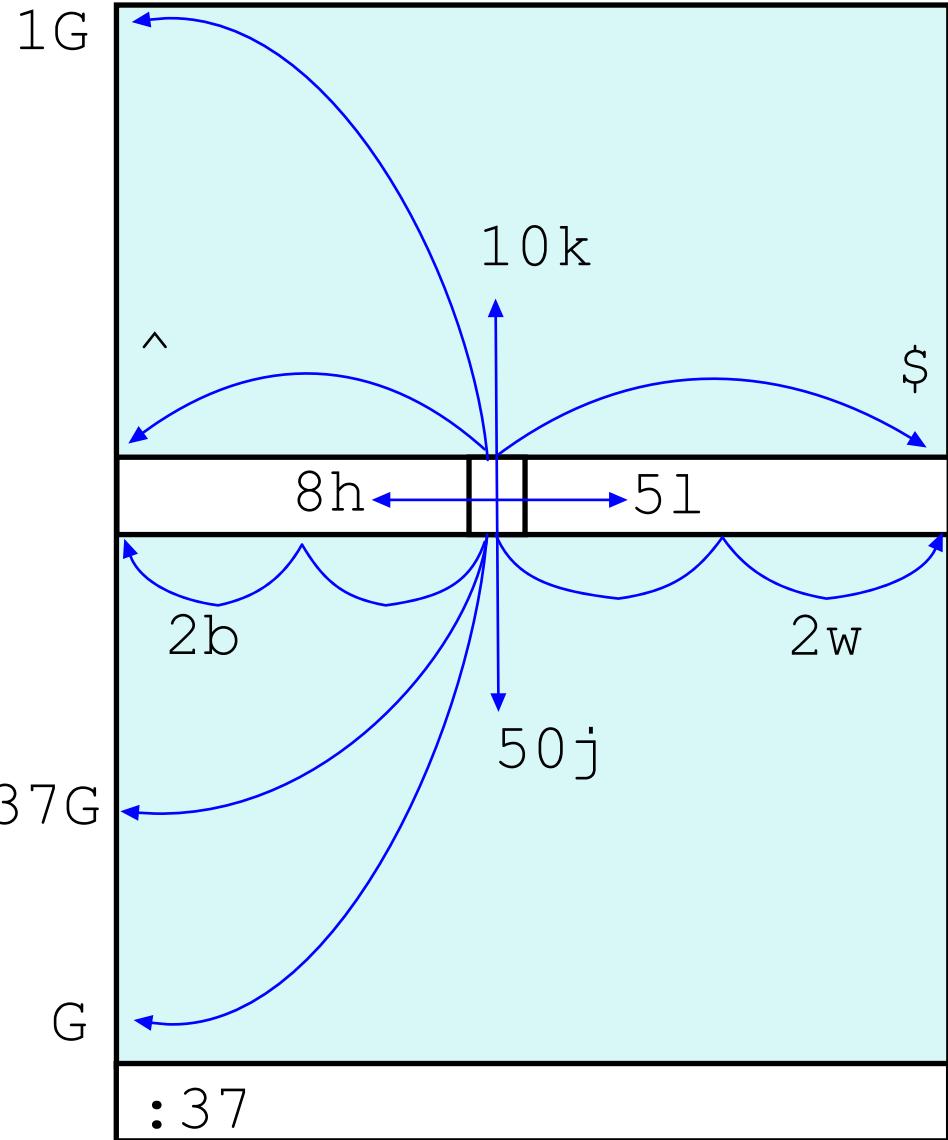
~/.exrc

```
:set showmode
:set number
:set autoindent
:set showmatch
:set tabstop=4
:set wrapmargin=40
:set ignorecase
:ab www http://www.qa.com
```

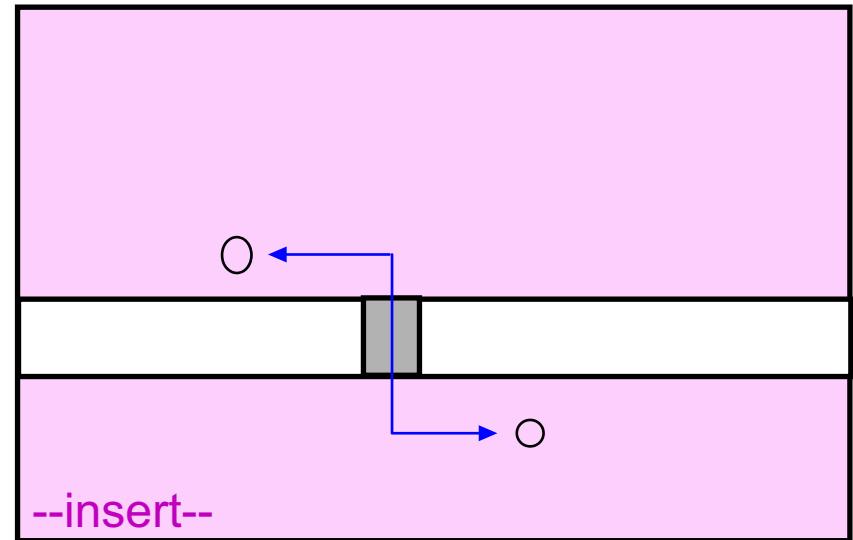
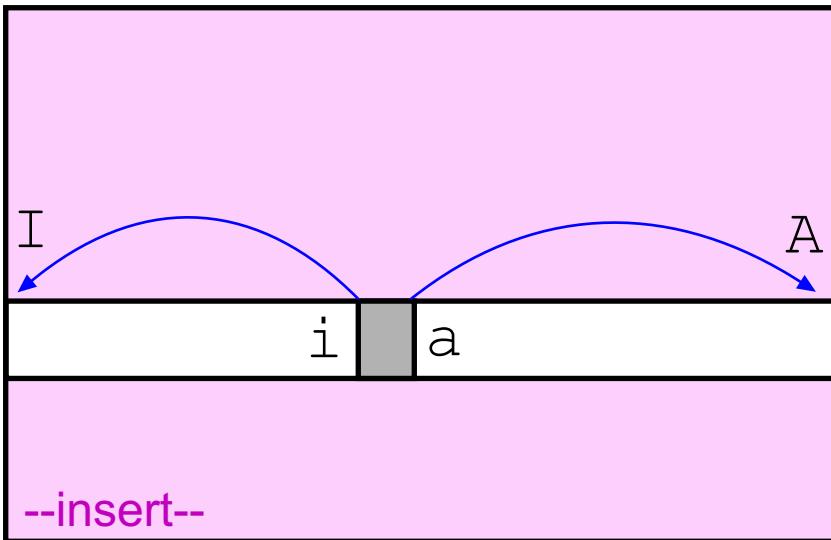
```
:set all # Display all settings
:set      # Displays enabled.
```



vi/vim - Navigation



vi/vim - Insert/Append/Open



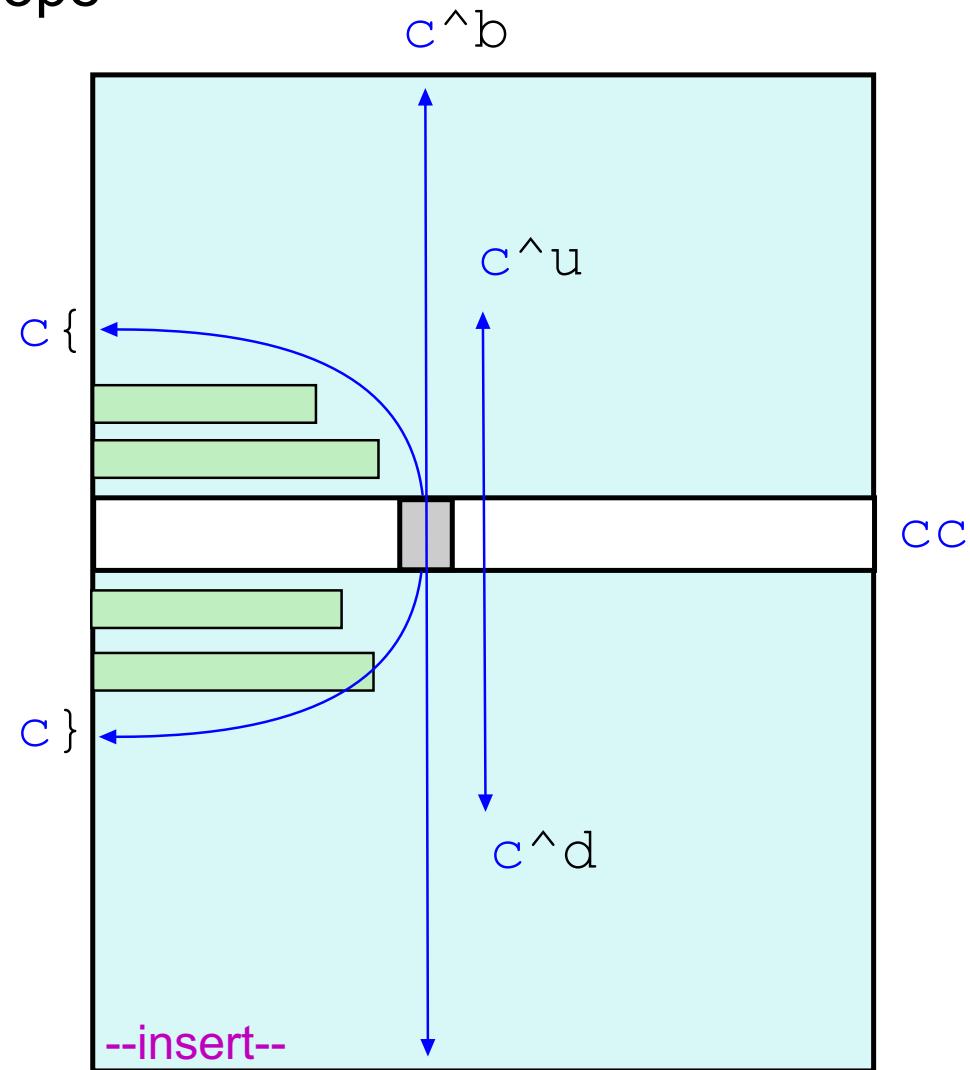
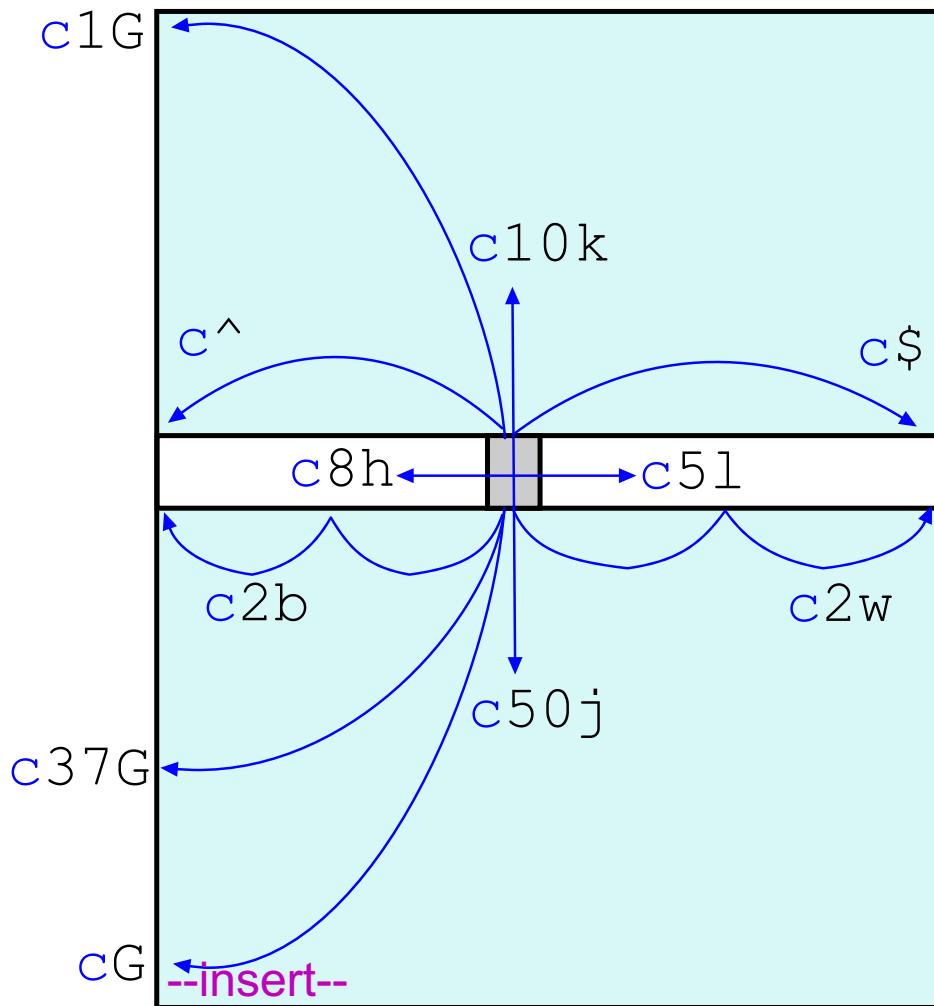
Backspace works!

Press **ESC** to go back into Command mode

:set showmode

vi/vim - Change Text

c<scope>



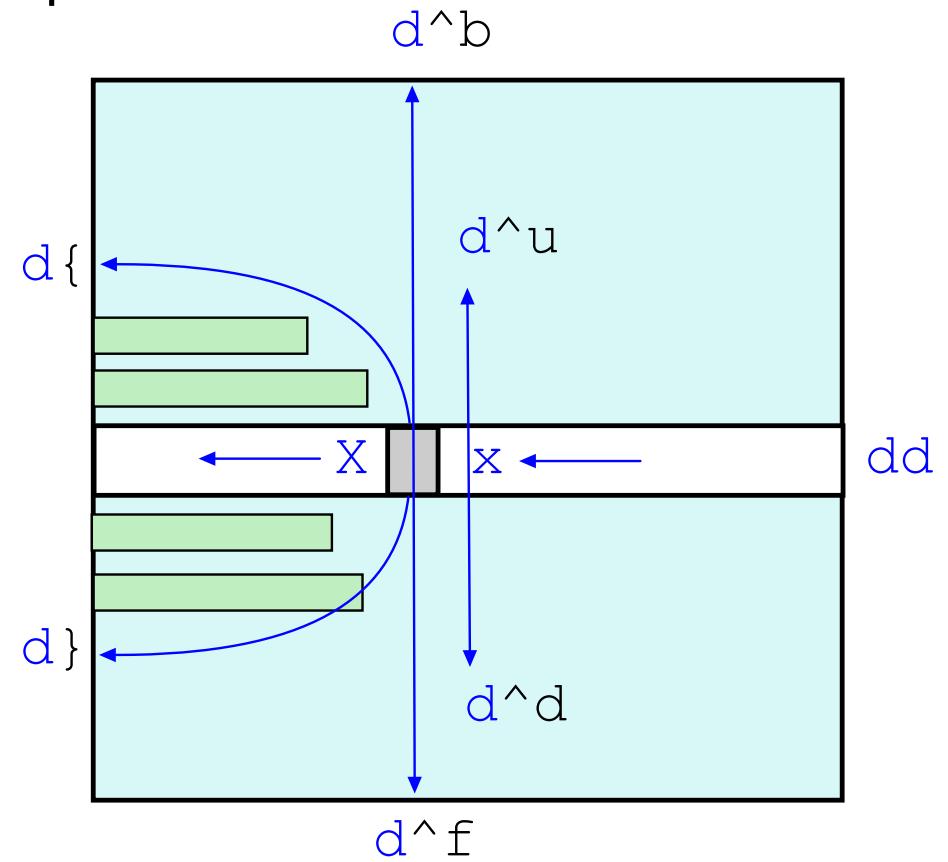
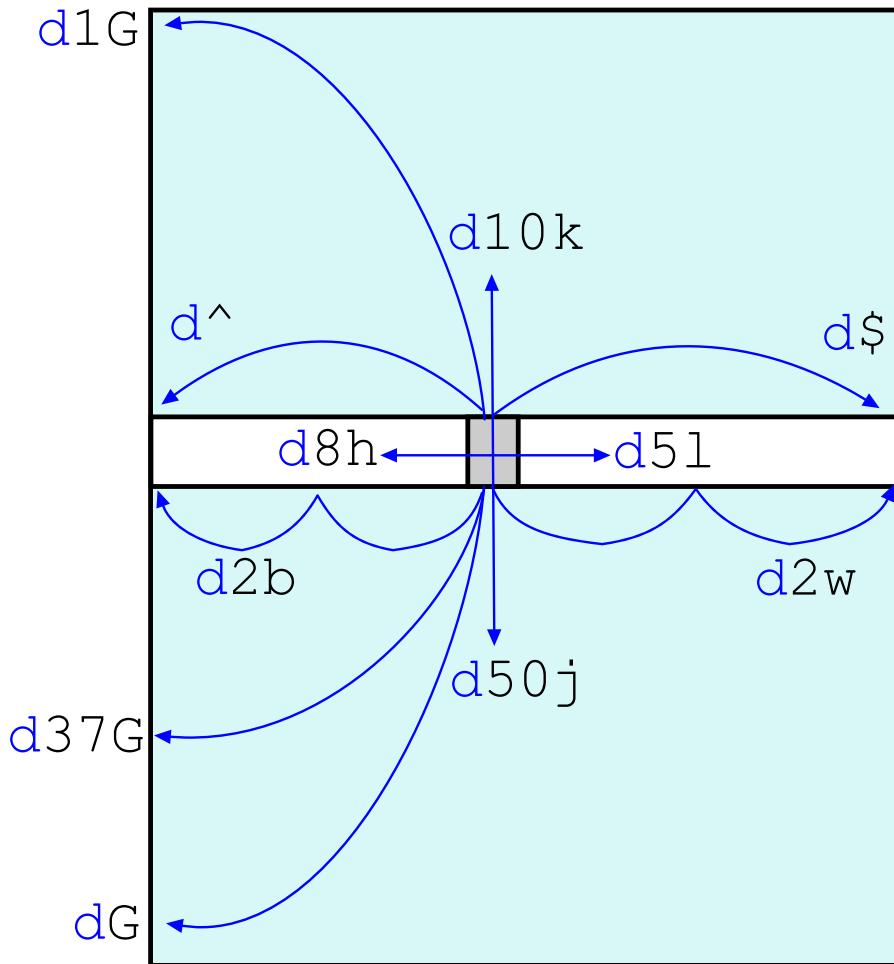
Backspace works!

Press **ESC** to go back into Command mode



vi/vim - Delete Text

d<scope>

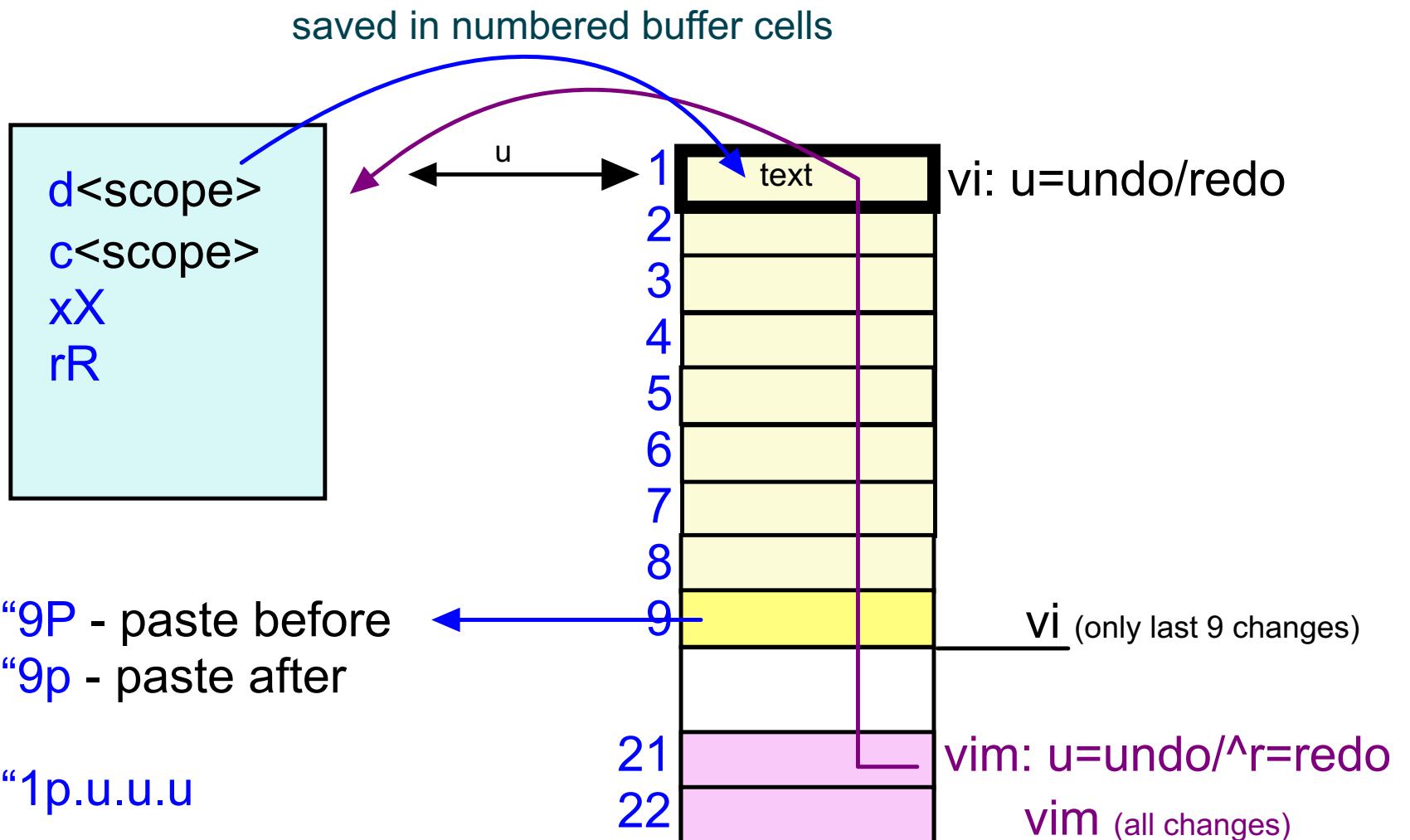


r = replace 1 char
R = overwrite/replace
~ = toggle cAsE

Press **ESC** to go back into Command mode

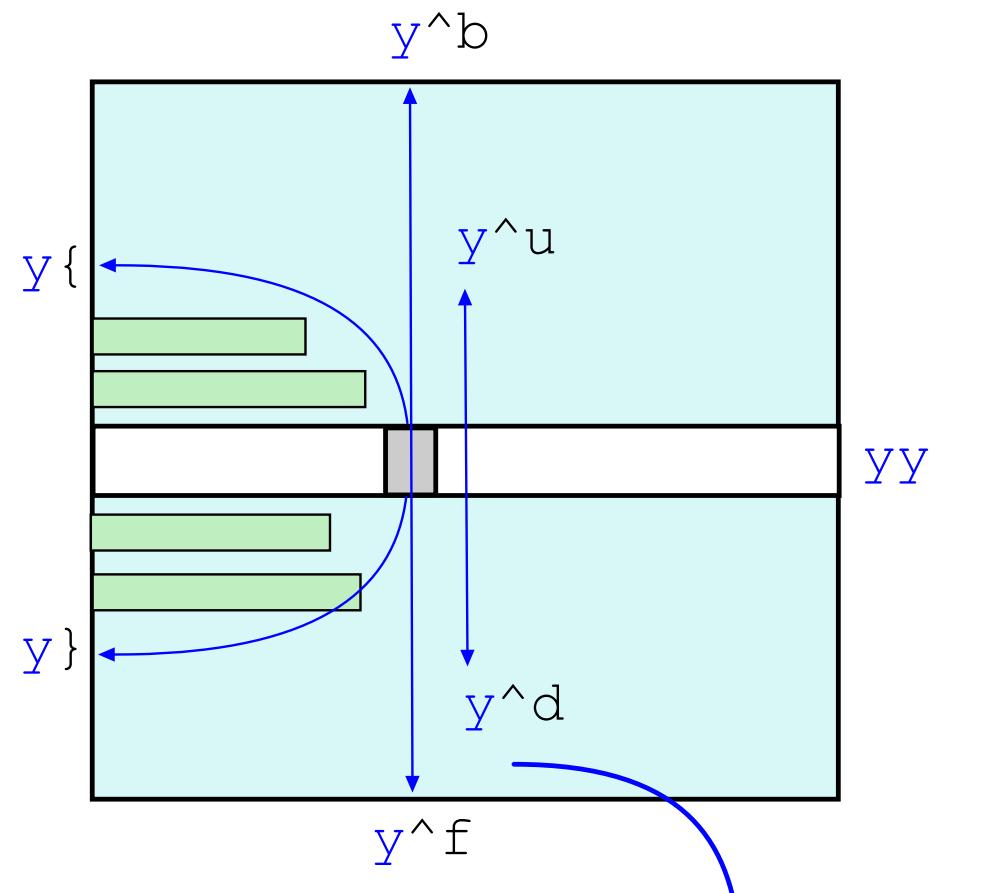
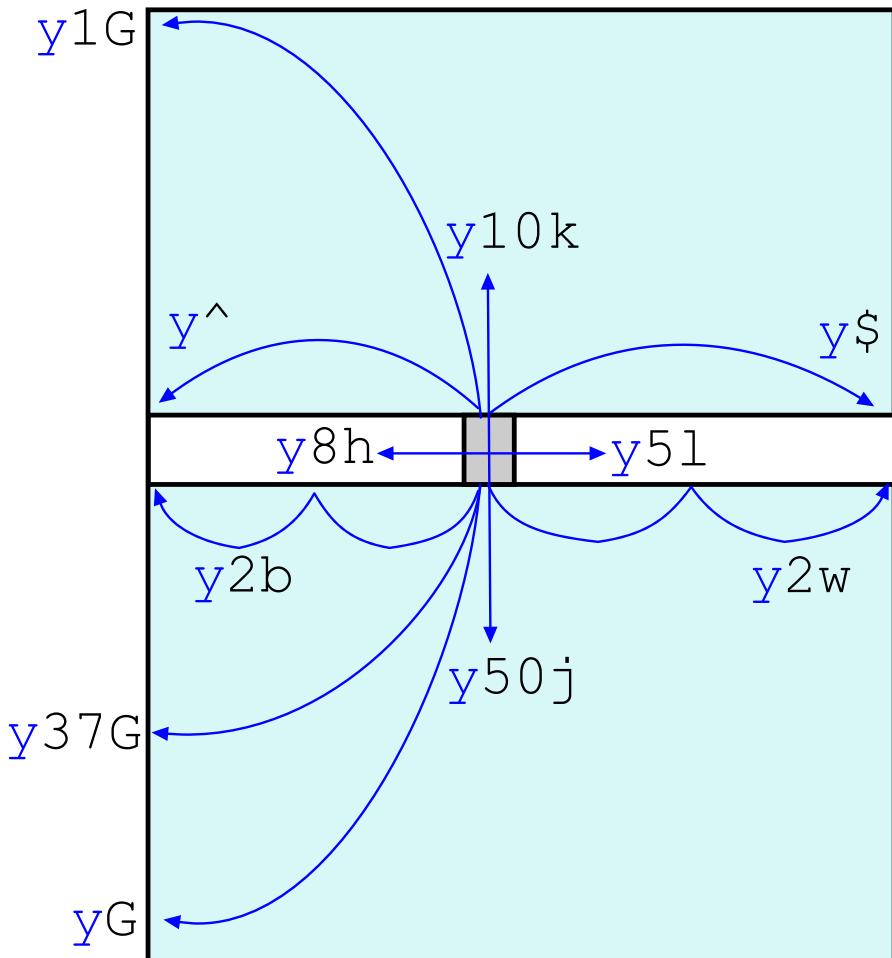


vi/vim - Cut & Paste/Undo



vi/vim - Yank & Paste

`y<scope>`



P - paste before
p - paste after

text text

copy
buffer

vi/vim - Copy, move and delete (ex mode)

Delete

```
:d          # delete current line  
:10d       # delete line 10  
:10,20d    # delete lines 10 to 20 inclusive  
:30,$d     # delete line 30 to last line inclusive
```

Copy

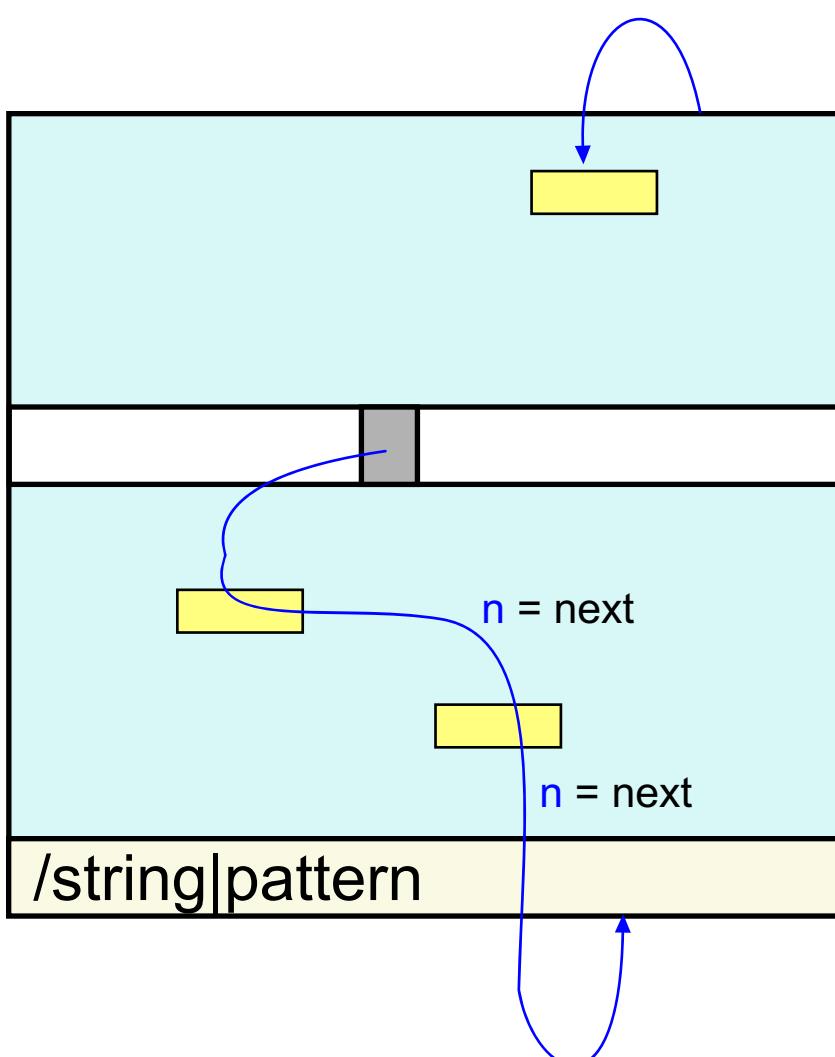
```
:co 40      # copy current line after line 40  
:40co 20    # copy line 40 after line 20  
:1,10co 120 # copy lines 1 to 10 after line 120  
:1co $      # copy line after last line
```

Move

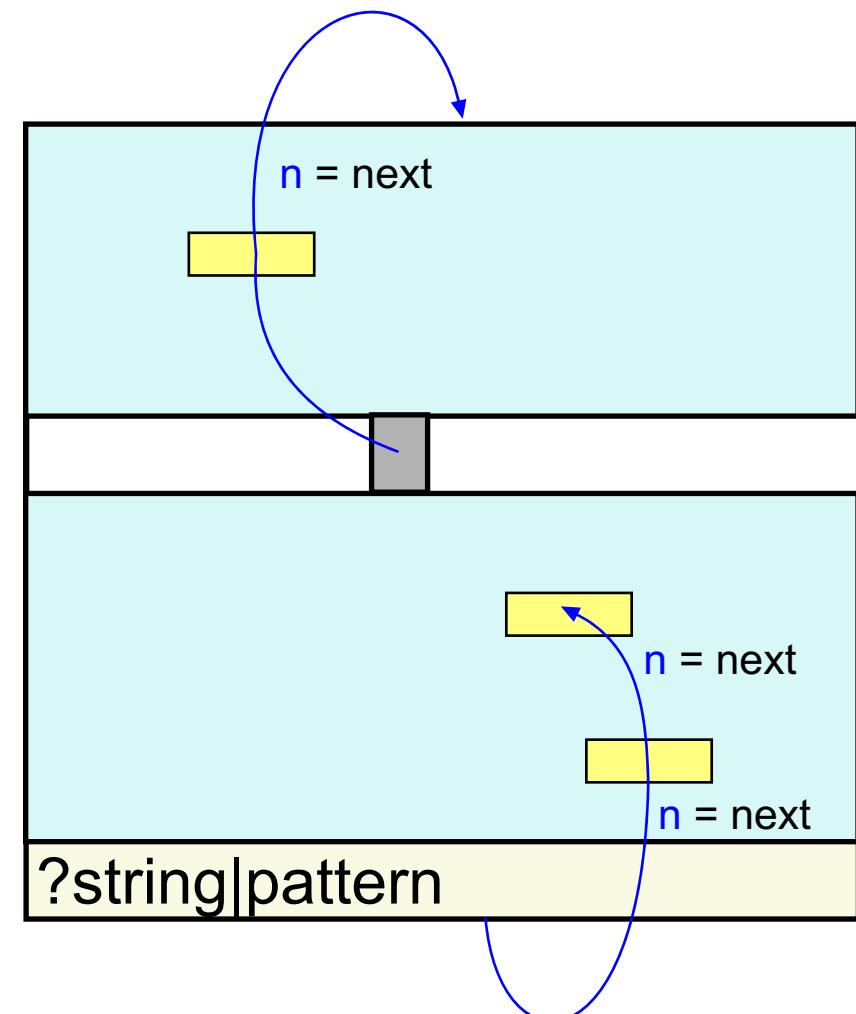
```
:mo10       # move current line after line 10  
:80mo 35    # move line 80 after line 35  
:1,10mo 90  # move lines 1 to 10 after line 90  
:$ mo 1     # move last line after line 1
```



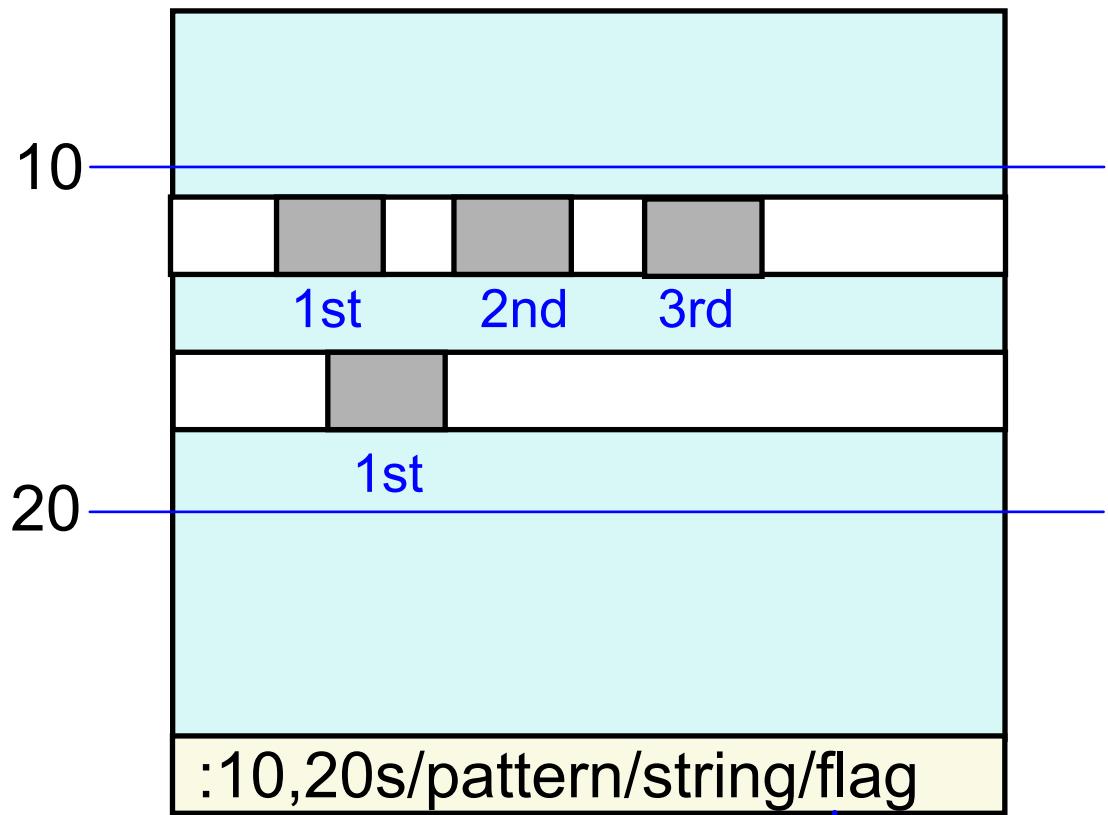
vi/vim - Fowards & Backwards Search



eg. /^[Tt]op [Ss]ecret\$



vi/vim - Substitution (ex mode)



Examples

:10,20s/^dDonald/Donald Cameron/1

:1,\$s/^dDonald/Sir & Cameron/1

:20,30s/^.*\$/ &/g

:20,30s/^/ /g

1-512 = occurrence
g = global/all occurrences
i = ignore case
c = confirm/prompt (vim only)



vi/vim - Quit, Read and Write (ex mode)

Quit

:q	# quit without saving
:q!	# quit without saving even if modified

Write

:w	# save file
:w filename	# save to filename
:wq	# save and quit
:wq!	# save and quite even if file is read only
:x	# quit and only save if modified (maintains time stamps)

Read

:r filename	# read and insert contents of file after current line
:56r filename	# read and insert contents of file after line 56

Run Commands

::! date	# run and display output of command
!! ls -l	# run and insert output of command
:56! who -r	# run and insert output of command after line 56



vi/vim - User Preferences (ex parameters)

\$HOME/.exrc

```
:set showmode|noshowmode          # enabled vi mode display
:set autoindent|noautoindent       # enable auto indent on next line
:set ignorecase|noignorecase       # enable ignore case
:set showmatch|noshowmatch         # enable show matching brackets
:set tabstop=4                      # set tabs to 4 spaces
:set wrapmargin=40                 # enable line wrap
:set number|nonumber                # display line numbers

:ab www  http://www.qa.com          # create abbreviation
```

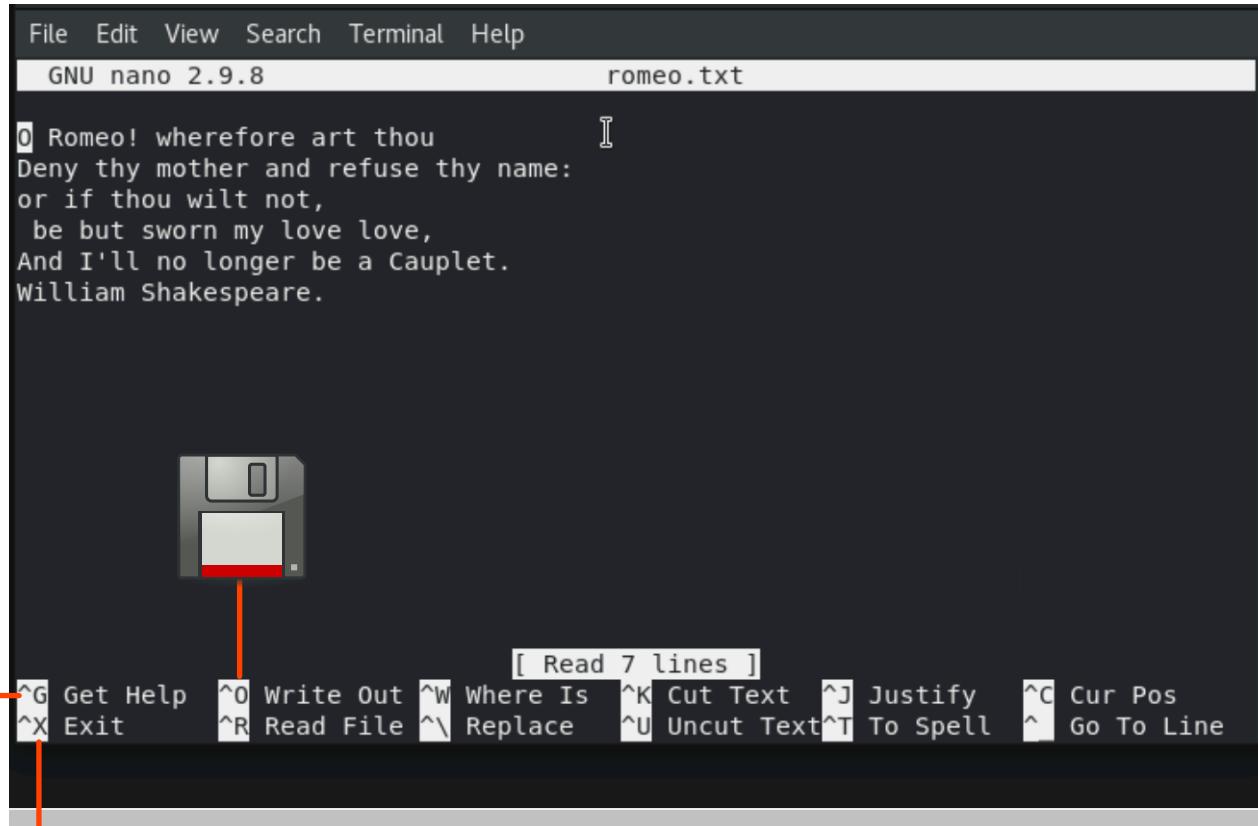
\$ vitutor
\$ vimtutor

vi/vim tutorial



Editing Files using Nano

\$ nano romeo.txt



Keyboard Shortcuts

Ctrl+S	Save
Ctrl+O	Save-as
Ctrl+R	Insert from file
Ctrl+X	Exit

Navigation

Arrow Keys

Ctrl+A	Start of line
Ctrl+E	End of line
Ctrl+Y	Page up
Ctrl+V	Page down

Editing

Ctrl+K	Cut line
Alt+6	Copy line
Ctrl+U	Paste buffer
Alt+U	Undo last action
Alt+E	Redo last action

Search

Ctrl+W	Forward search
Ctrl+Q	Backward search

Shell Metacharacters

* ? [-] #
\$ \ ' " ` &

Interprets line first →

\$ command {+-options} arg1 arg2 arg3

launches
command
last

Shell Metacharacters

eg. Globbing characters: ? [abc] [!abc] [a-z] *

Quoting characters: \ ' " `



Shell Globbing chars = wildcards!

(Purpose in life - match existing file/Dir names)

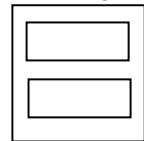
Globbing characters: Single-char wilcards: ? [abc] [!abc] [a-z]
Multi-Char wildcard: *

\$ cat reportA reportB reportC ... reportZ > final.report

\$ cat report? > final.report

\$ cat report[A-Z] > final.report

\$ rm file[A-Z][0-9]



\$ echo ?????.txt

\$ echo *.* c* D*

\$ echo *.*?? *.[a-z][a-z][a-z]

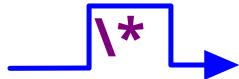
\$ ls -l .[!]*

\$ touch new.[A-Z]



Shell Quoting/Escape Metachars

Backslash



Escapes all Shell metachars x1

Single Quotes



Escapes all Shell metachars xn

Double Quotes



Escapes all Shell metachars
except \$ \ ``

Command Substitution

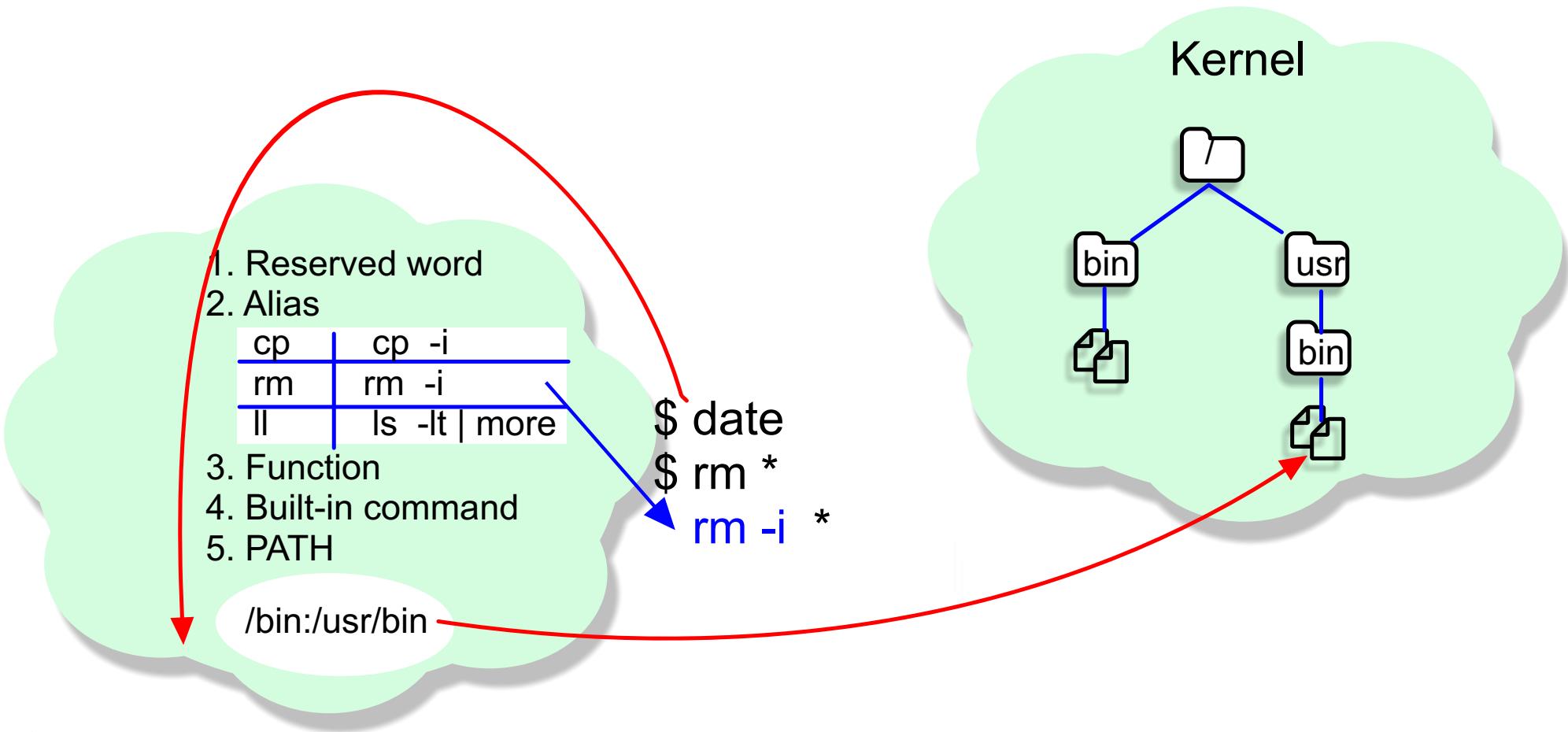


Back Quotes\Ticks `subcommand` \$(subcommand)

```
$ echo "time and date is `date`"  
$ echo "Current run level is $(runlevel)"  
$ ls -l > log.`date +%j`  
$ ls -l > log.$(date +%d-%m-%Y)
```



Shell Command Alias Substitution (bash)



Example

```

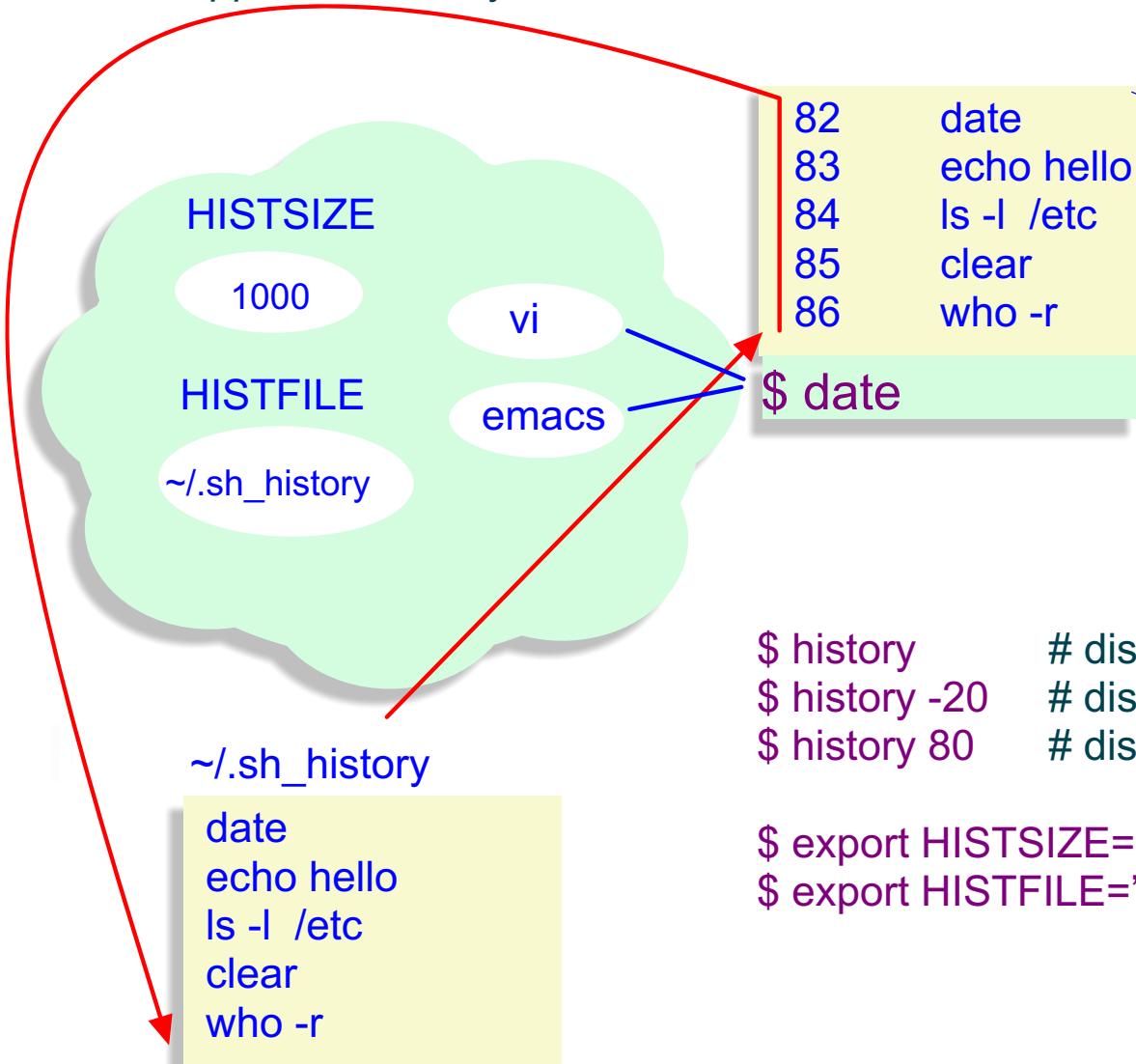
alias rm='/bin/rm -i'
alias cp='/bin/cp -i'
alias mv='/bin/mv -i'
alias c=clear
alias ll='/bin/ls -lt | /bin/more'
  
```

```

alias      # display all aliases
alias ll   # display ll alias
unalias ll # remove alias
  
```

Shell History (ksh)

Also append to history file



`$ history` # display last 16 commands
`$ history -20` # display last 20 commands
`$ history 80` # display history commands from number 80.

`$ export HISTSIZE=500`
`$ export HISTFILE='~/.sh_history_$(basename $(tty))'`

`$ set -o vi` # enable vi commands on history
`$ set +o vi` # disable vi commands on history

Shell History (bash)

Append/Overwrite history file
when session is ended

HISTSIZE
200

HISTFILE
~/.bash_history

HISTFILESIZE
1500

vi
emacs

~/.bash_history

```
date
echo hello
ls -l /etc
clear
who -r
```

\$ shopt -s histappend

82	date	! 82
83	echo hello	! e
84	ls -l /etc	! -3
85	clear	
86	who -r	!!

\$ date

<esc> + vi commands (hjkl,i,a,r,x,cw,dw)
or
<arrow keys> + backspace

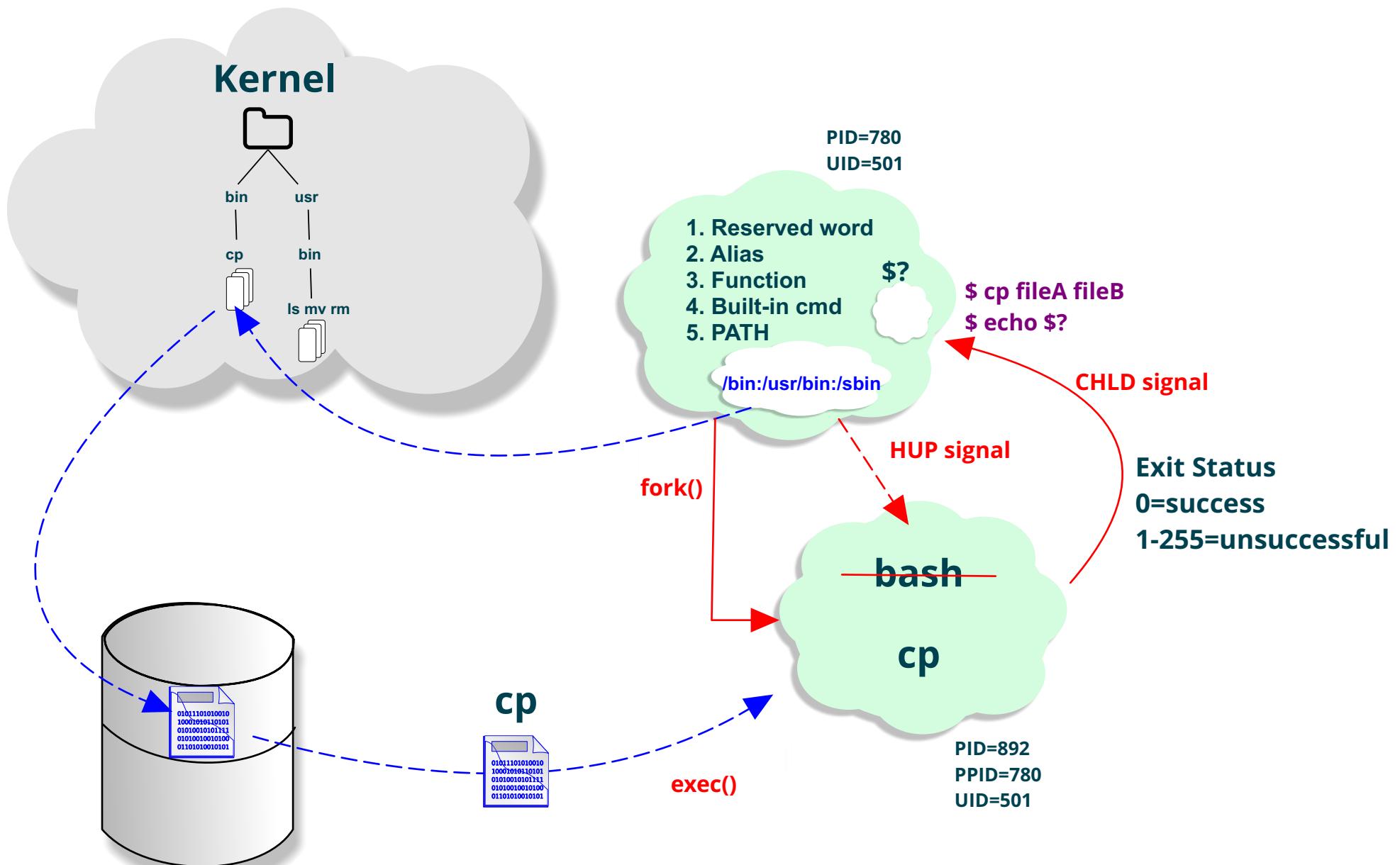
\$ history # display history list
\$ history 30 # display last 30 commands

```
$ export HISTSIZE=500
$ export HISTFILE='~/.bash_history_$(basename $(tty))'
$ export HISTFILESIZE=1500
$ export HISTIGNORE='cd:date:ls:echo:history'
```

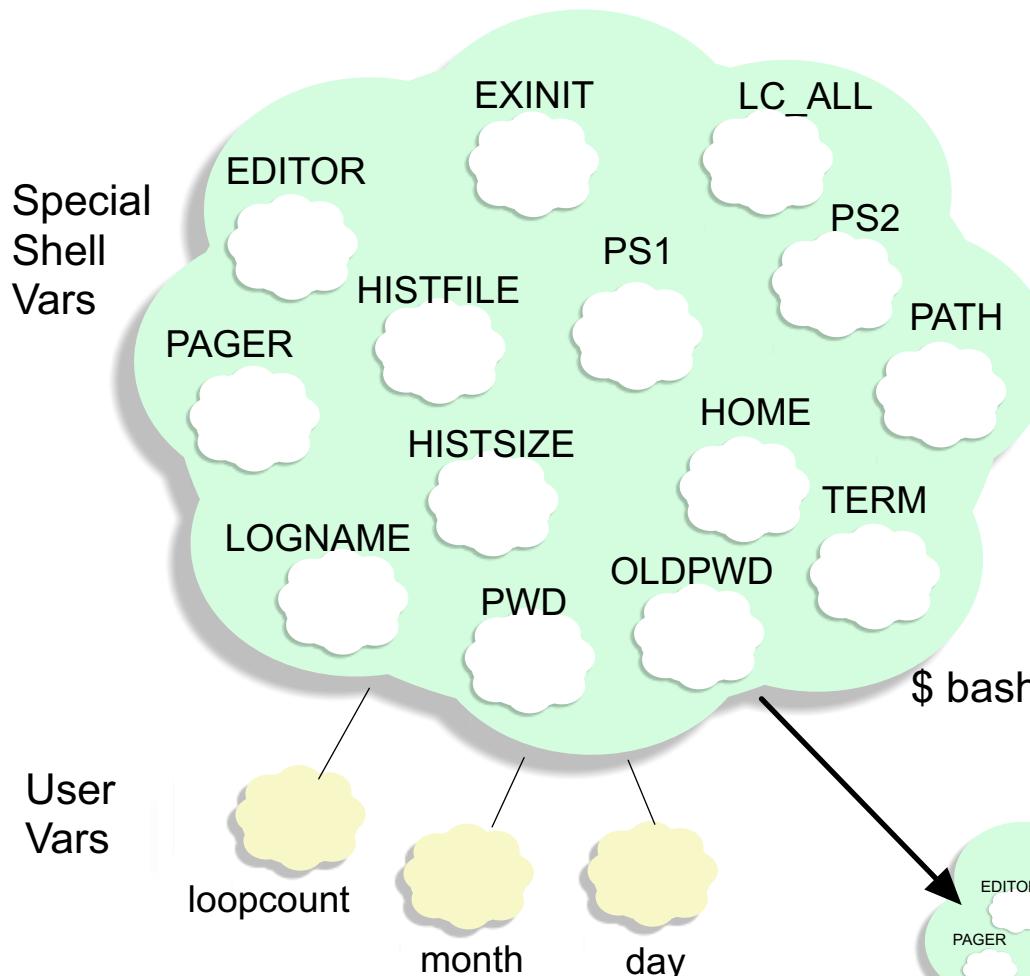
\$ set -o vi # enable vi commands on history
\$ set +o vi # disable vi commands on history



Shell Environment - the life of UNIX/Linux Process

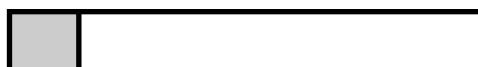


Shell Variables (ksh)



Variable Naming Convention

255 chars



1st char cannot be 0-9 \$#!?-*@

Create user variables

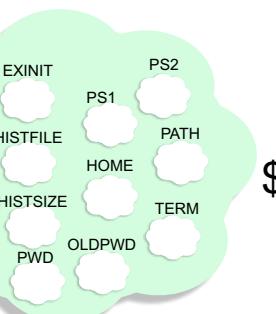
```
$ count=100
$ a=10 b=20 c=30
$ name="Donald Cameron"
$ day=$(date +%d)
$ month=$(date +%m)
```

Reset/unset user variables

```
$ count=42
$ unset a b c day month
```

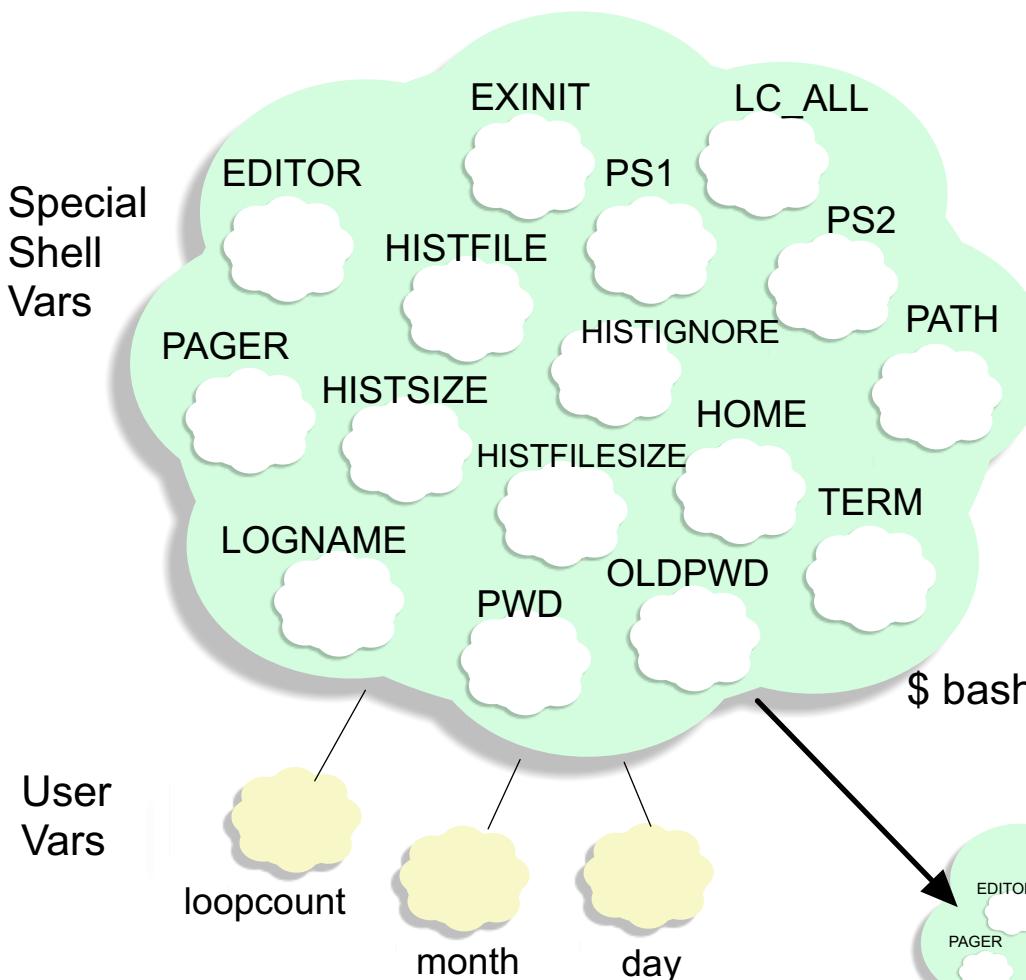
Display user variables

```
$ set
$ echo "$day $month $PATH"
```



\$

Shell Variables (bash)



Variable Naming Convention

255 chars



1st char cannot be 0-9 \$#!?-*@

Create user variables

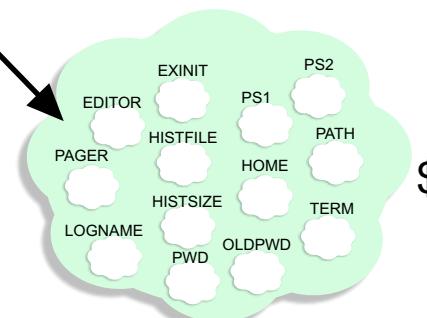
```
$ count=100
$ a=10 b=20 c=30
$ name="Donald Cameron"
$ day=$(date +%d)
$ month=$(date +%m)
```

Reset/unset user variables

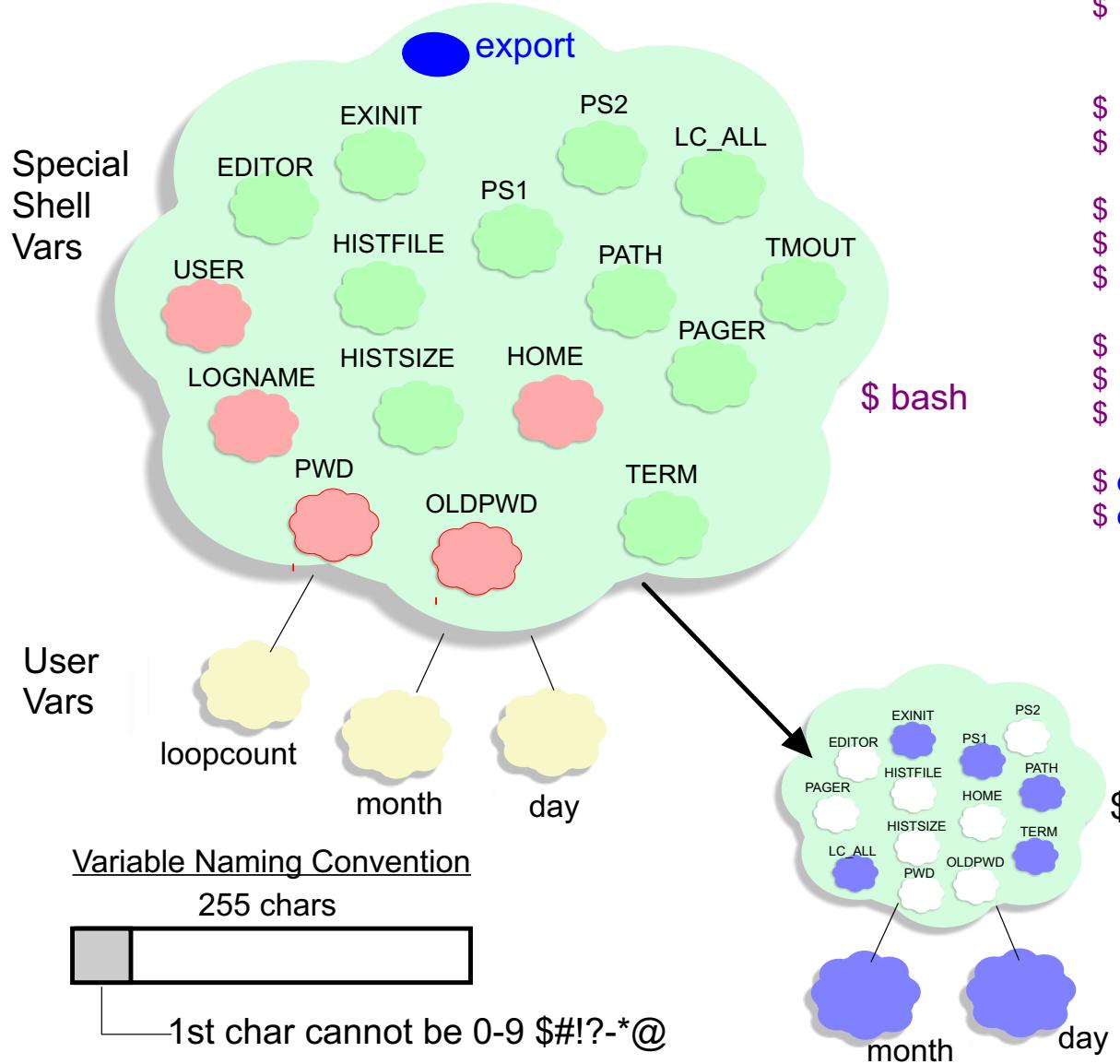
```
$ count=42
$ unset a b c day month
```

Display user variables

```
$ set
$ echo "$day $month $PATH"
```



LOCAL Shell Variables (ksh)



```

$ PATH="${PATH}:${HOME}/bin:${HOME}/scripts::"
$ PS1="${LOGNAME}@$(uname -n):${PWD}[\!]$ "
$ PS2="cont> "

$ LC_ALL=en_GB.utf8
$ locale -a

$ EXINIT=set showmode number|ab www www.qa.com"
$ TERM="vt100"
$ find / -name 'terminfo' 2> /dev/null

$ HISTSIZE=100
$ HISTFILE=~/.sh_history
$ TMOUT=600

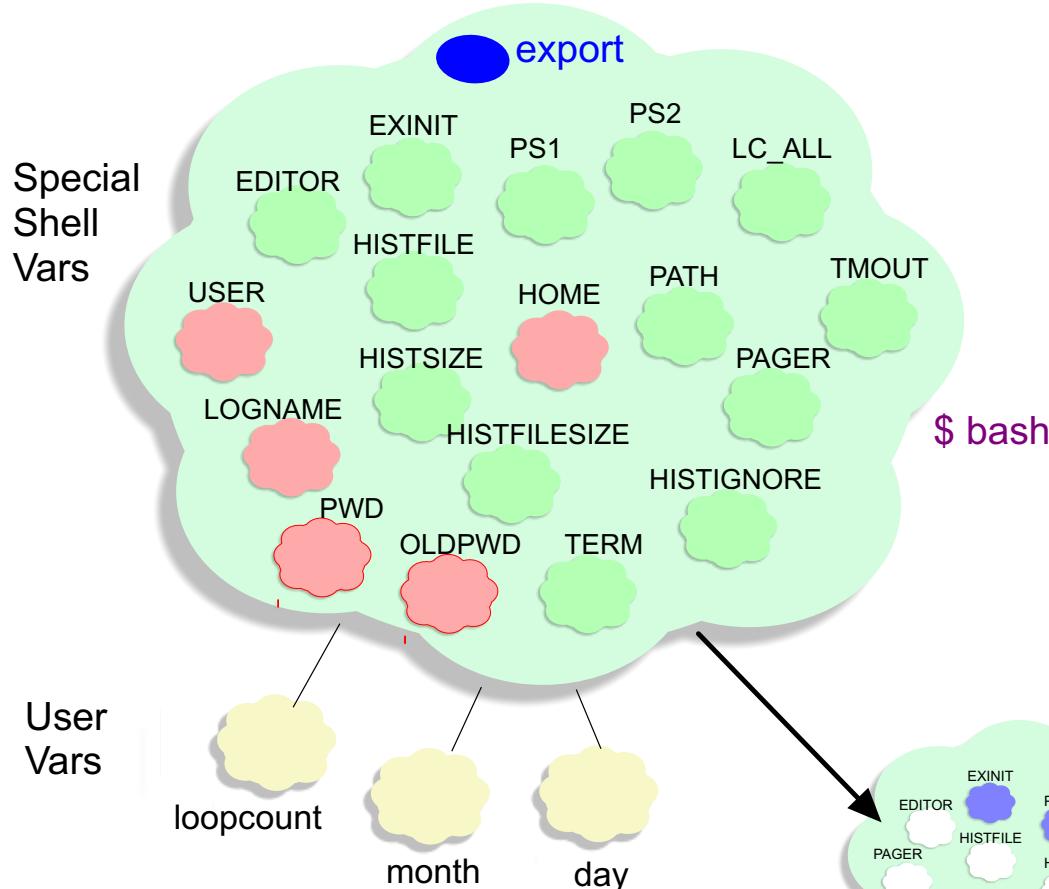
$ LPDEST="laserjet"
$ PAGER="/bin/more"
$ EDITOR="/bin/vi"

$ export PATH PS1 LC_ALL EXINIT TERM day month
$ export PATH="${PATH}:$HOME/bin"

```

[see: man ksh]

LOCAL Shell Variables (bash)

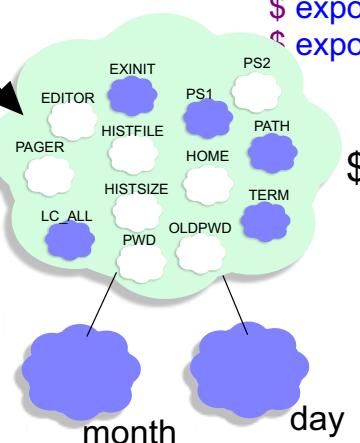


Variable Naming Convention

255 chars



1st char cannot be 0-9 \$#!?-*@



```
$ PATH="${PATH}:~/.bin:~/.scripts:."
$ PS1="${LOGNAME}@$(uname -n).${PWD}$
$ PS1="\e[0;31m\u@\e[0;33m\h\e[m:\W[\!]$ "
$ PS1="\u@\h:\W[\!]$ "
```

man bash
/PROMPTING

```
$ PS2="cont>
$ LC_ALL=en_GB.utf8"
```

locale -a

```
$ EXINIT="set showmode number|ab www www.qa.com"
$ TERM="vt100"
$ HISTSIZE=100
$ HISTFILE="${HOME}/.bash_history"
$ HISTFILE="${HOME}/.bash_history_$(basename $tty)"
$ HISTFILESIZE=1500
$ HISTIGNORE="cd:echo:ls:date:pwd"
```

find / -name 'terminfo' 2> /dev/null

```
$ TMOUT=600
$ LPDEST="laserjet"
$ EDITOR="/bin/vi"
$ PAGER="/bin/more"
```

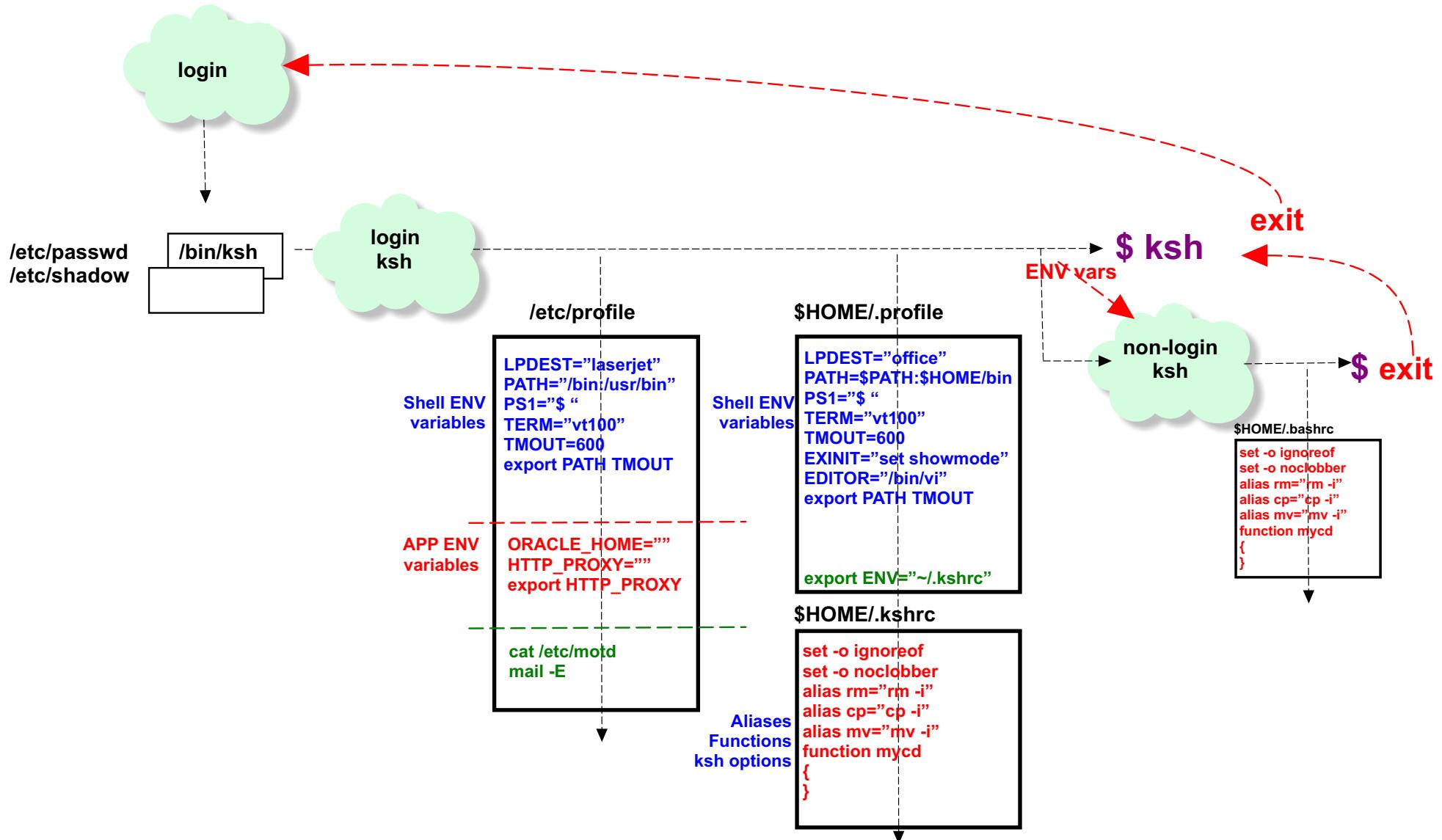
```
$ export PATH PS1 LC_ALL EXINIT TERM day month
$ export PATH="${PATH}:~/.bin"
```

[see: man bash]

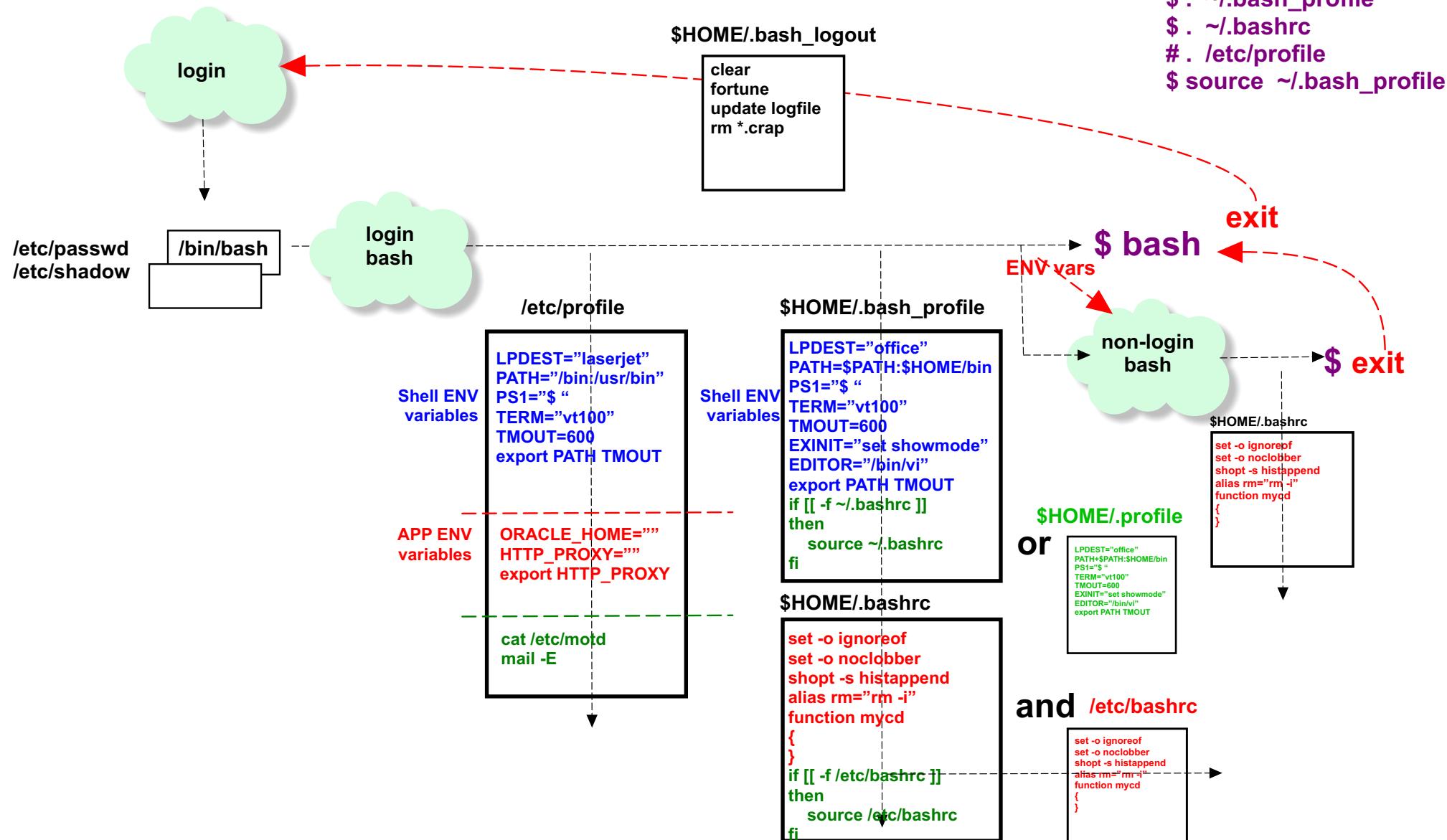
ksh Initialisation Files

Test/Source=reread

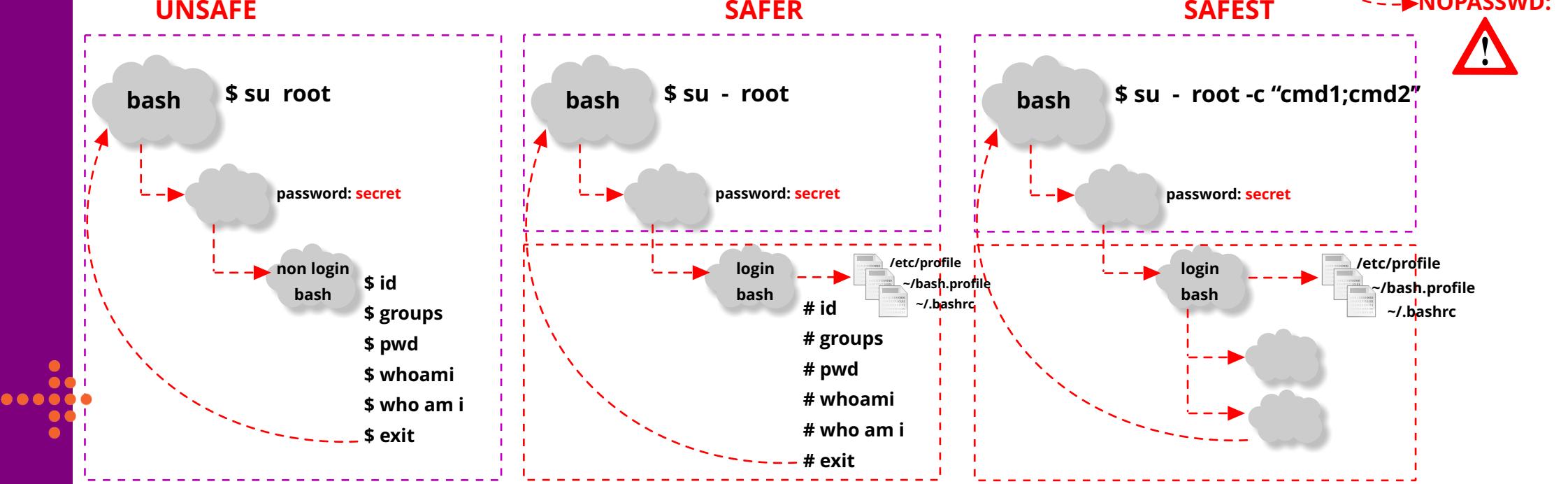
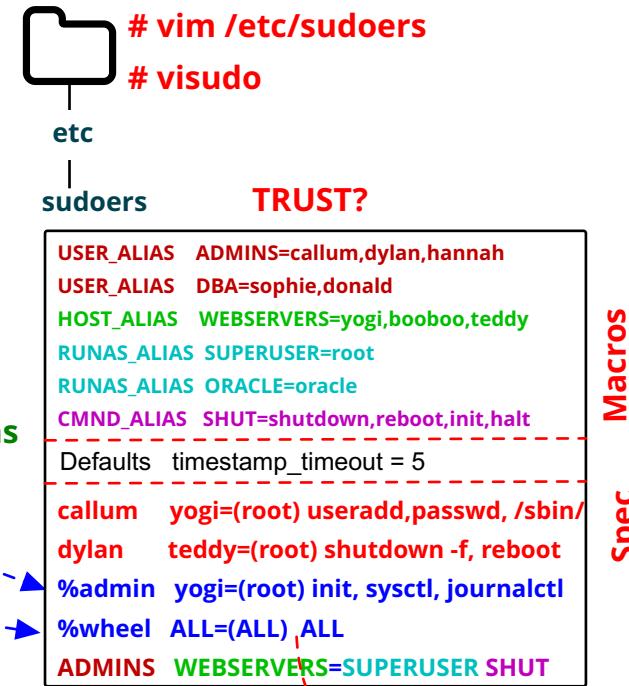
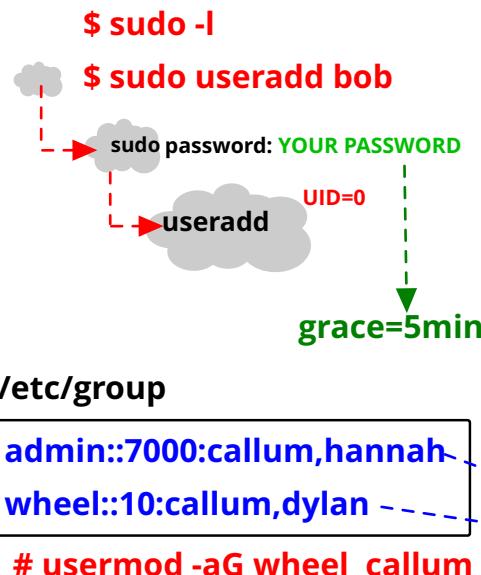
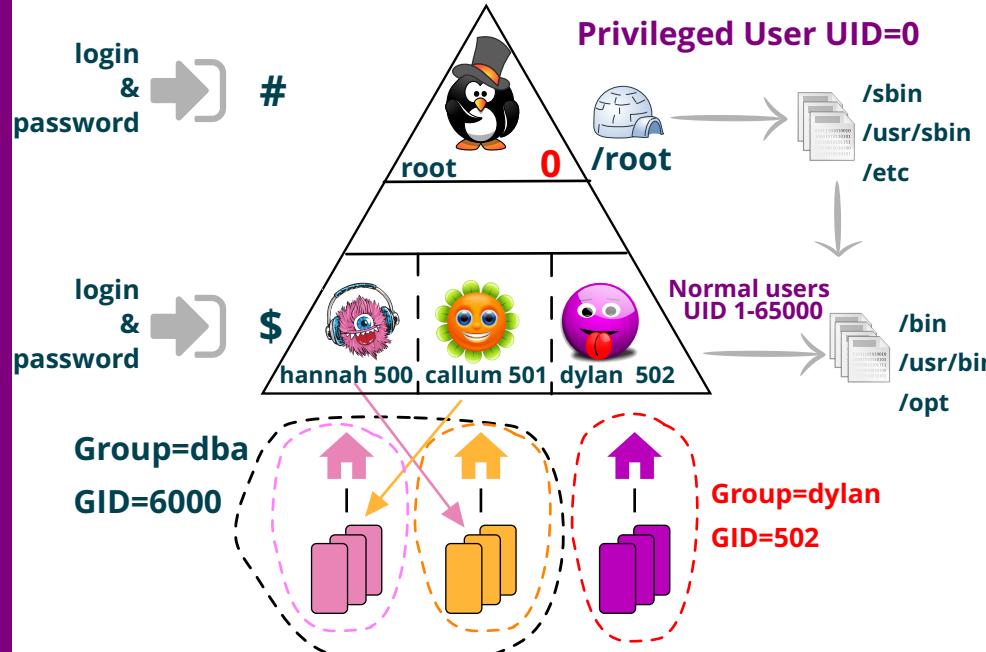
```
$ . ~/.profile
$ . ~/.kshrc
# . /etc/profile
```



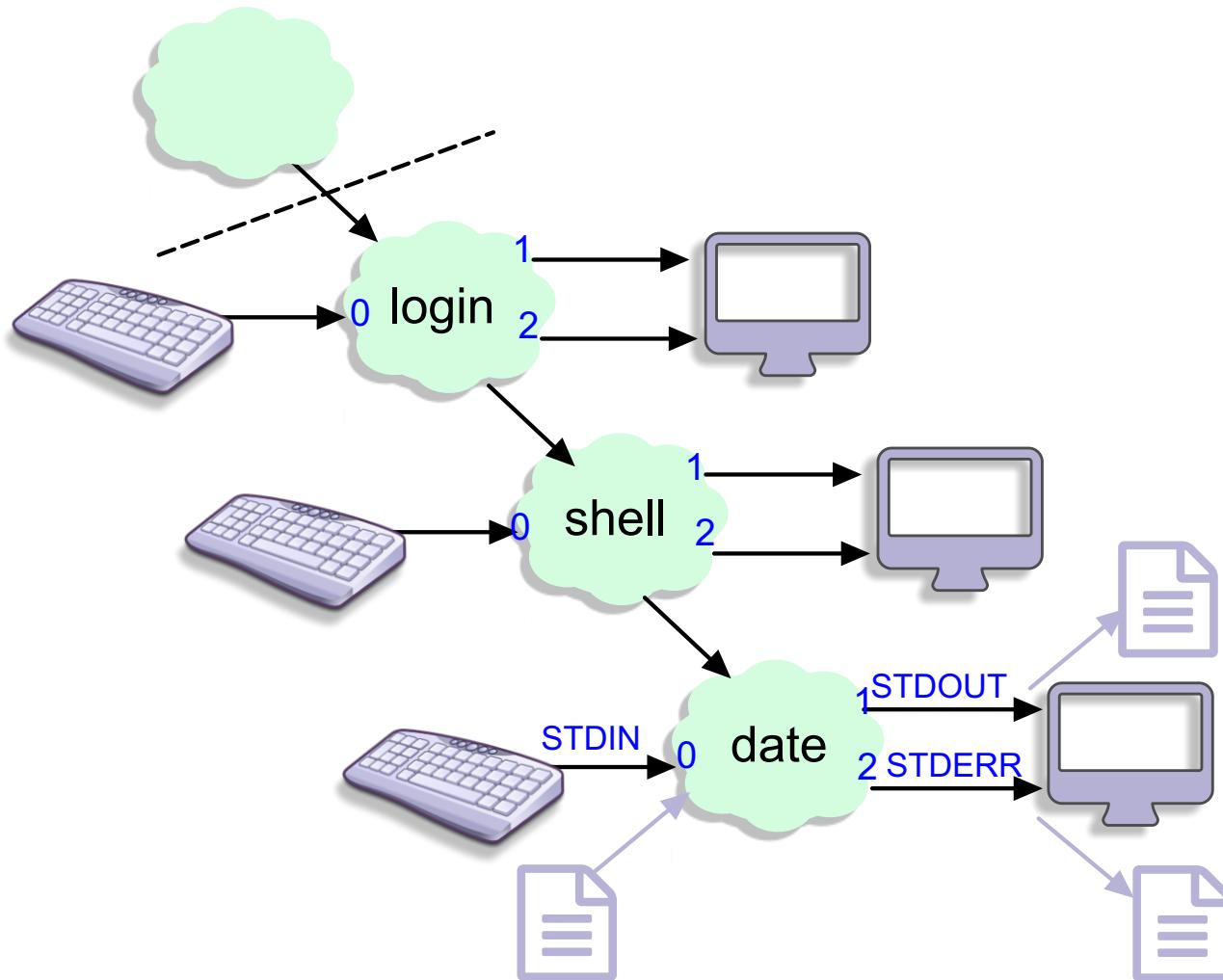
Bash Initialisation Files



Who is the Boss (using su and sudo)



Shell Redirection/Data Streams



Shell Redirection

0 STDIN 0< 0<<

1 STDOUT 1> 1>>
 overwrite append

2 STDERR 2> 2>> 2>&1 2>|
 overwrite append

~/.bashrc
~/.kshrc

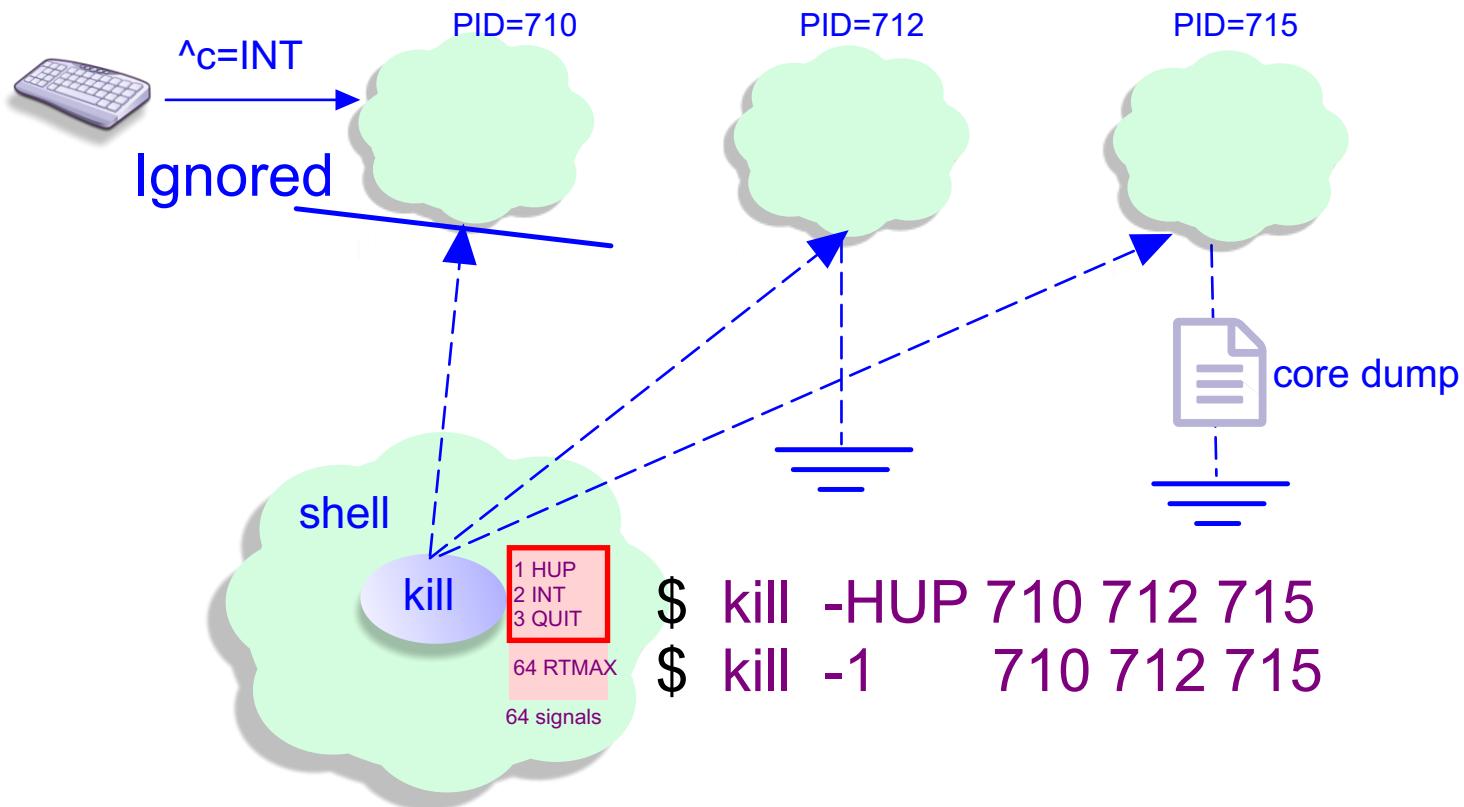
set -o noclobber

>|

2>|

override
noclobber

Shell Signals



```
$ kill -I
$ kill -I HUP
```

```
$ kill -I 1
```

```
$ man 7 signal
```

```
$ man -s3head signal
```

```
$ man -5 signal
```

```
# Linux
```

```
# Solaris
```

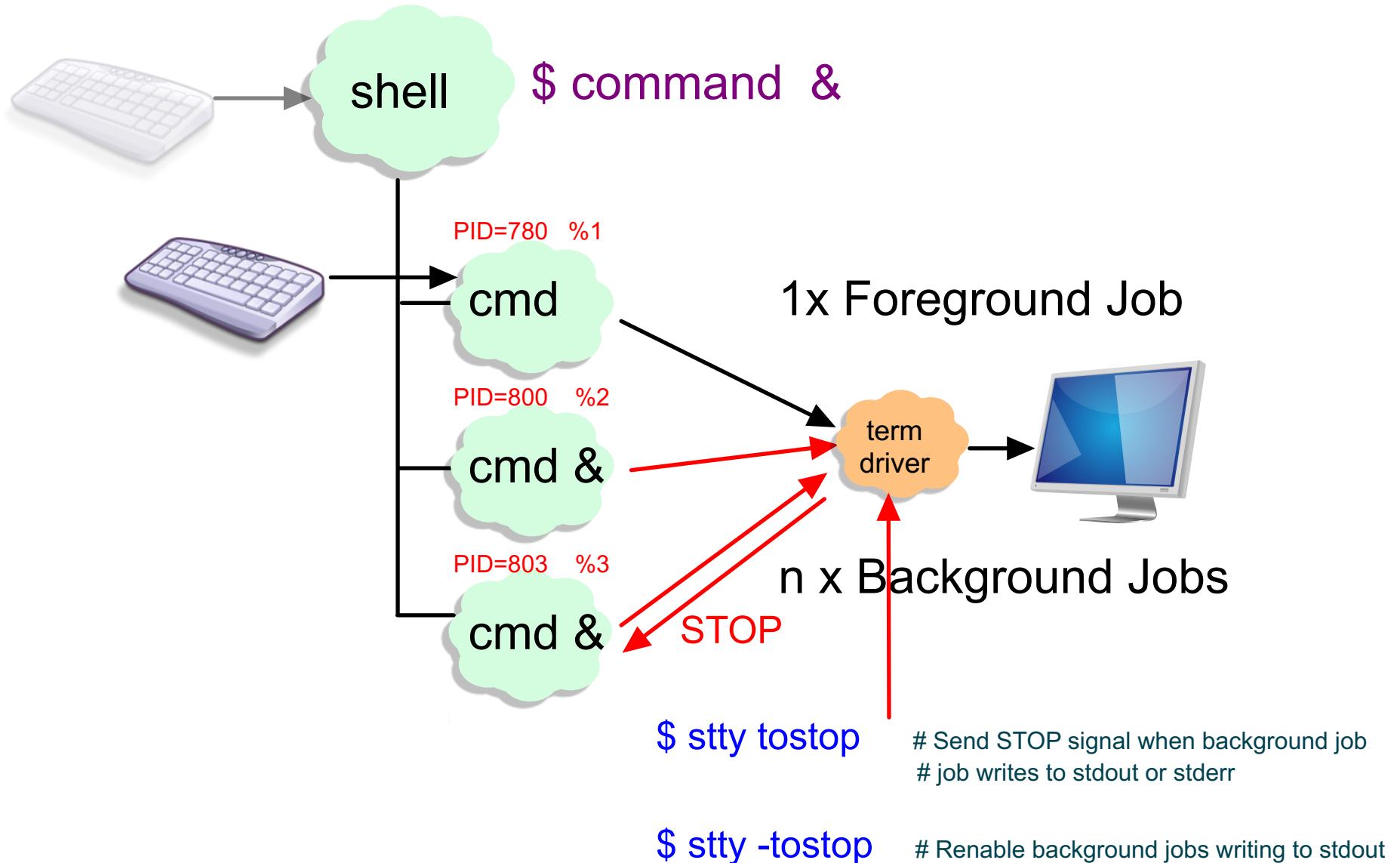
```
# AIX
```

Default

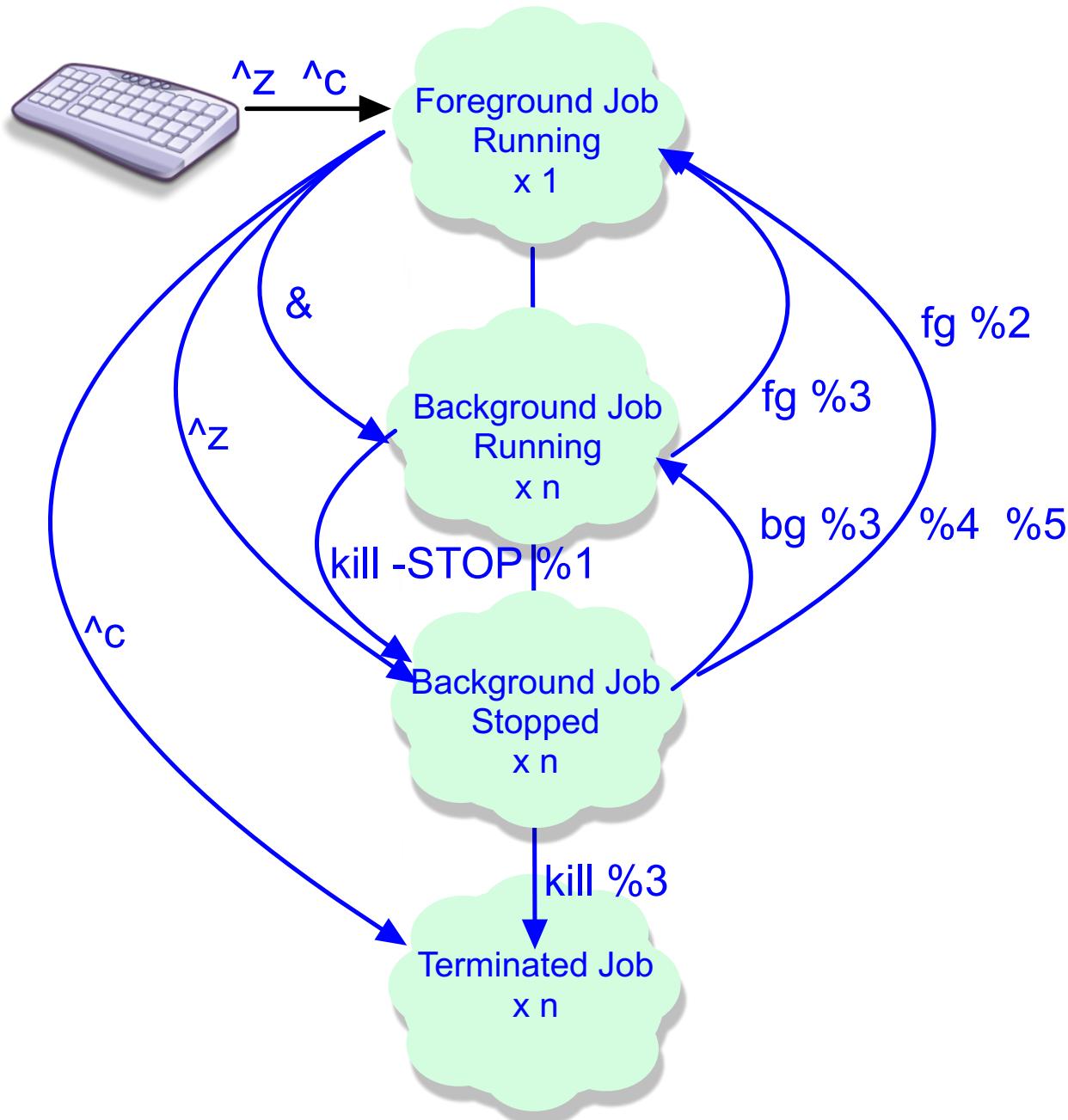
2nd	1 HUP	Hangup & terminate
3rd	2 INT C	Interrupt
3rd	3 QUIT $^{\backslash}Q$	Quit with core dump
last	9 KILL	COD Headshot
1st	15 TERM	Please terminate
	23 STOP	Stop Running
	24 TSTP Z	Suspend
	25 CONT	Continue running



Shell Job Control

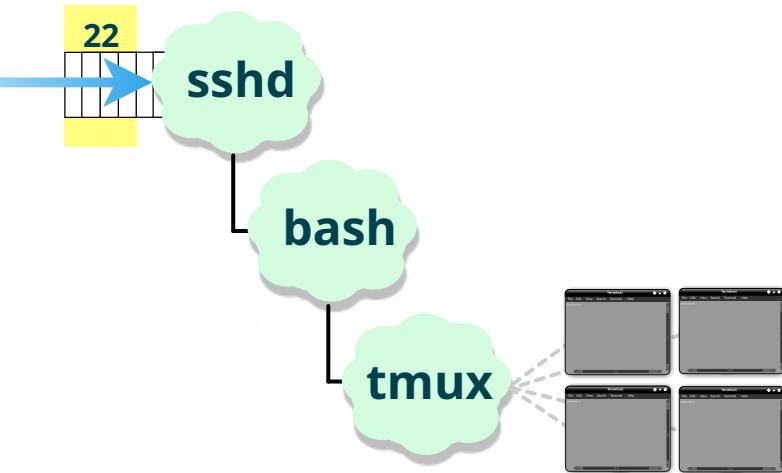


Shell Job Control



Terminal Multiplexers

tmux

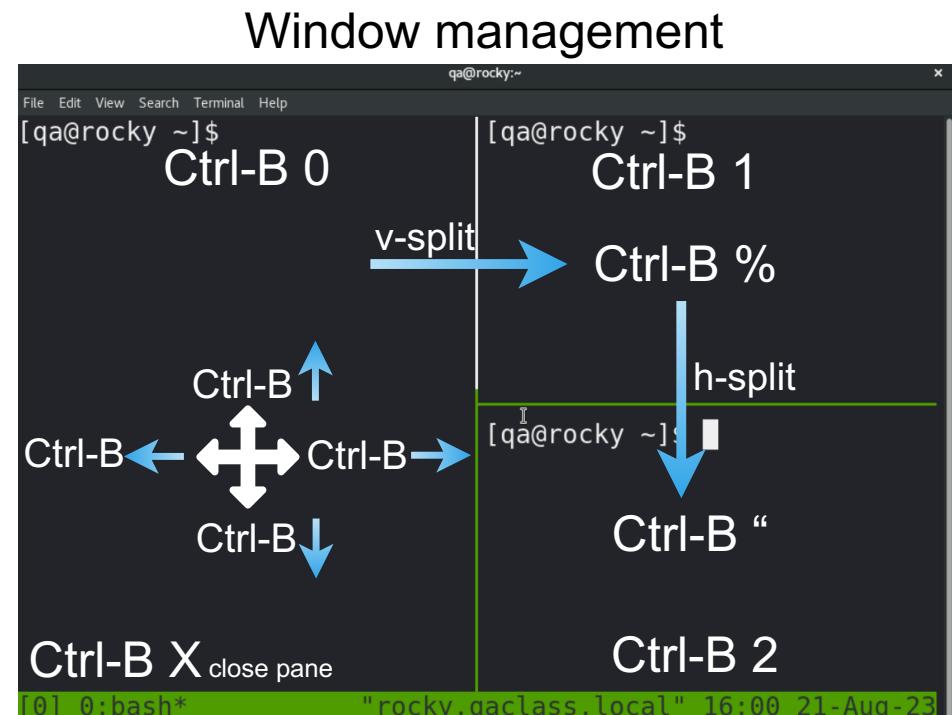


- Alternative to shell job control
- Creates Multiple Pseudo terminals on single connection
- Decouples programs from main terminal
- Detach tmux from terminal
- Programs continue to run in background
- Rettach to same or different terminal

```
$ sudo dnf -y install tmux
$ sudo apt-get install tmux
$ tmux
```

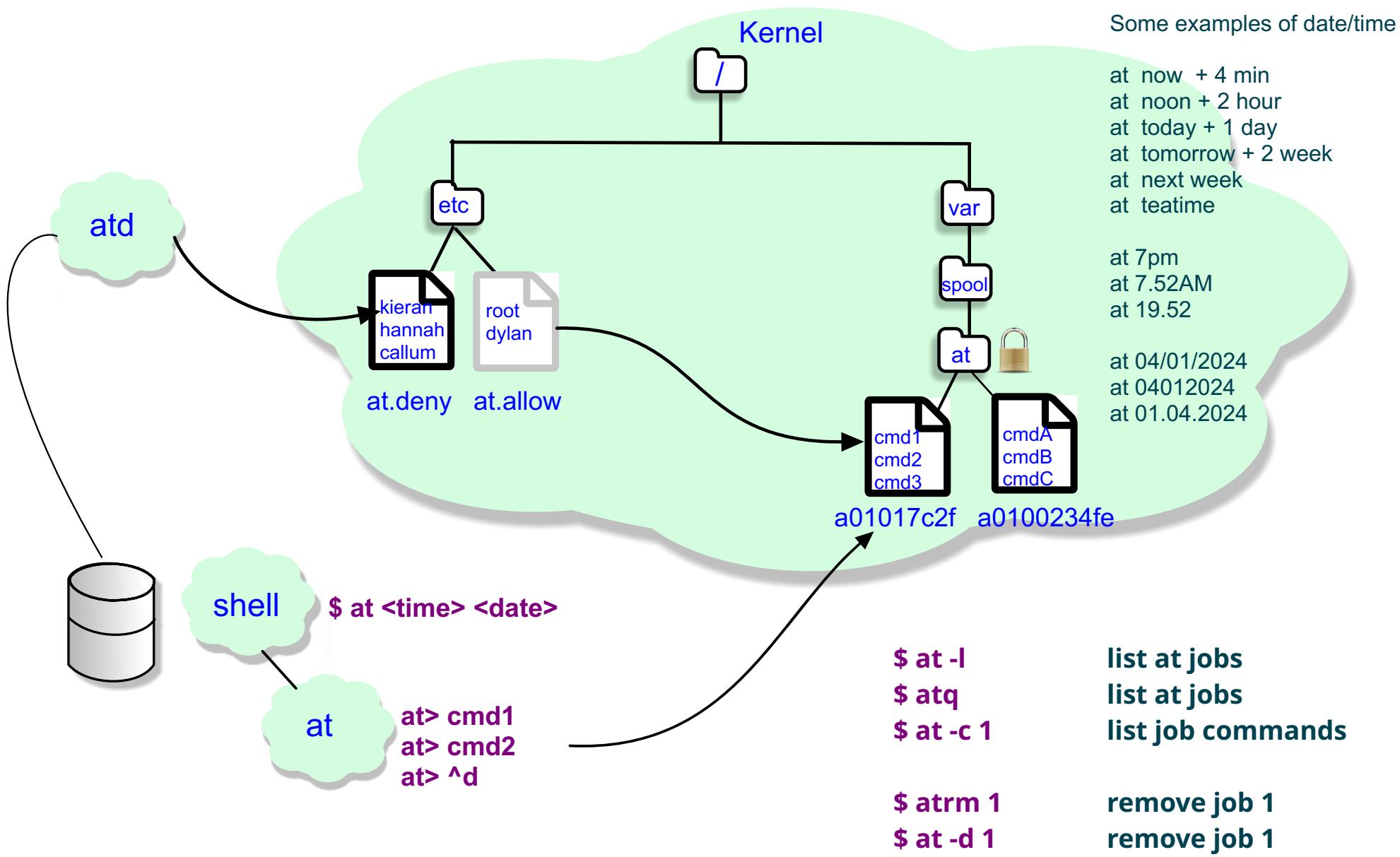
Detach
[detached (from session 0)]

Rettach
[reattach session 0)]



Mouse control: **Ctrl-B** : [and enter] set -g mouse

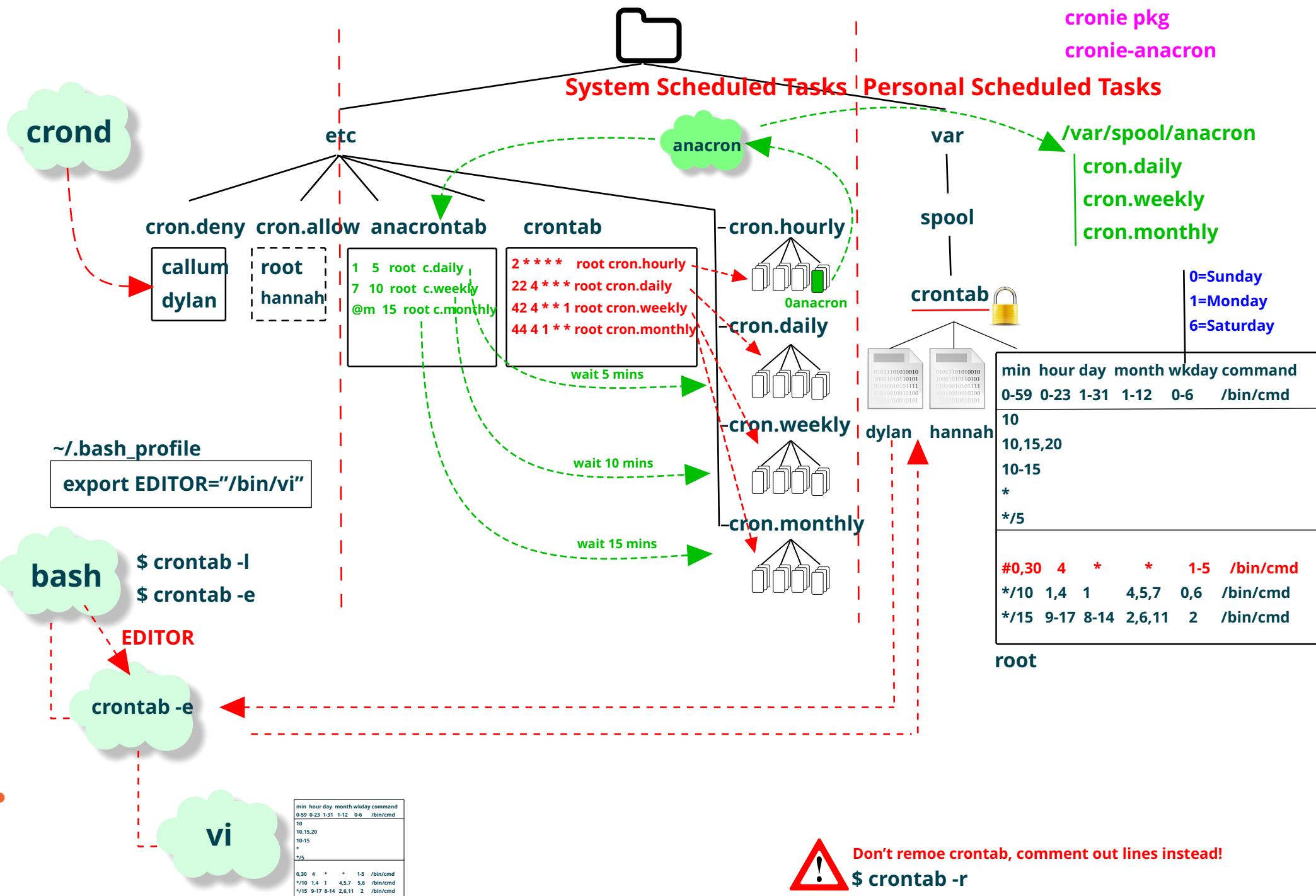
Scheduling Jobs using atd



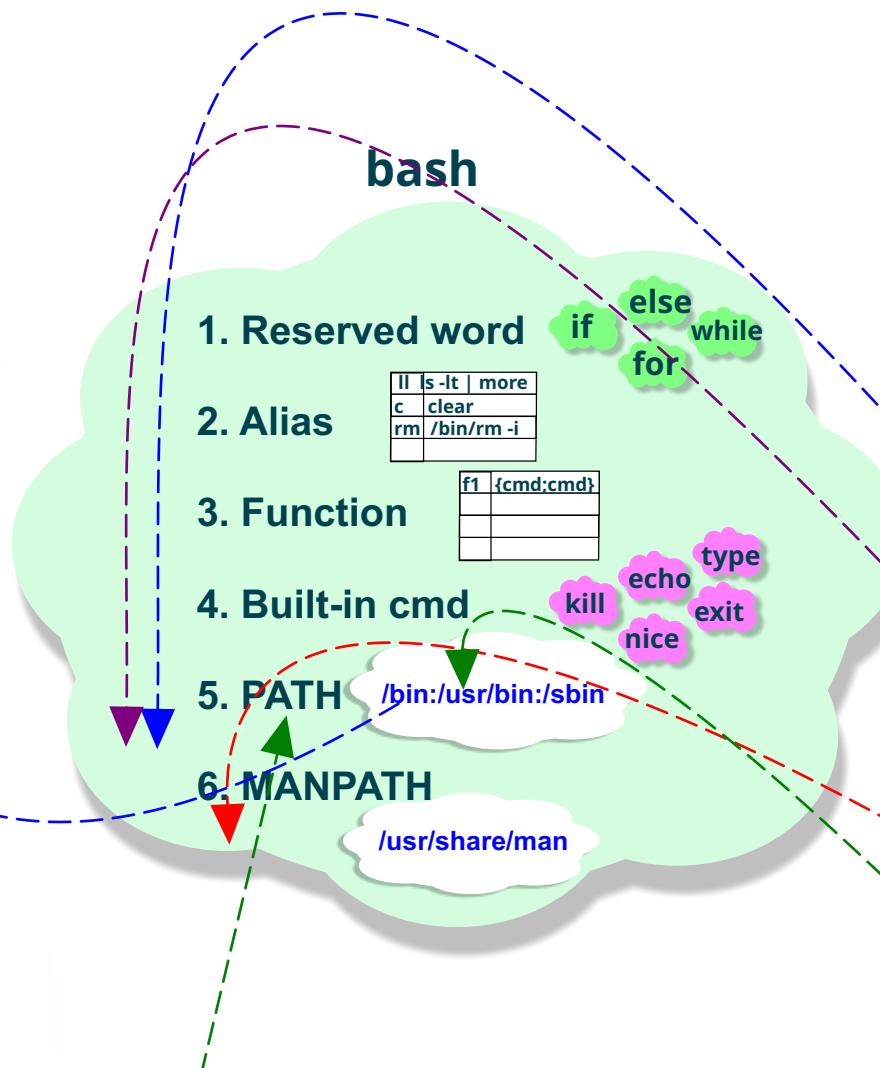
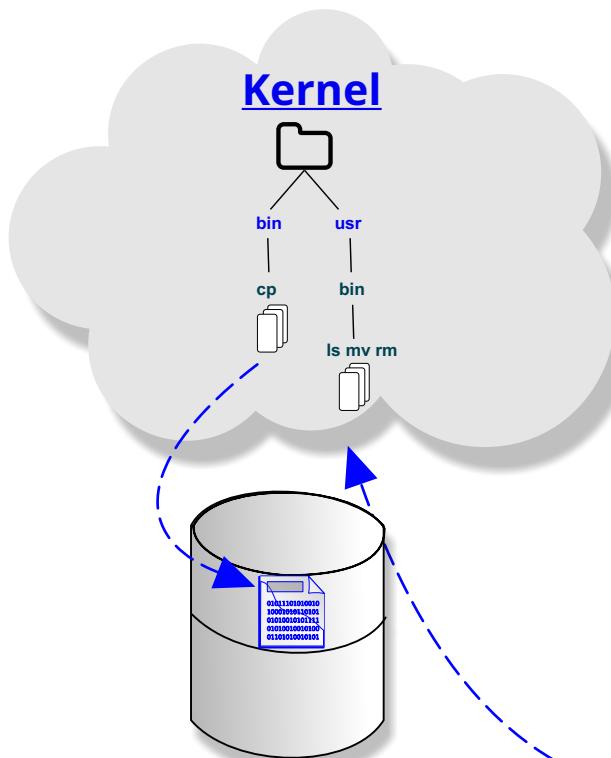
See man pages for `<time>` and `<date>`

Scheduling Tasks using crontab

```
$ rpm -qa | grep 'cron'  
crontab pkg  
cronie pkg  
cronie-anacron
```



Shell is PATH aware!

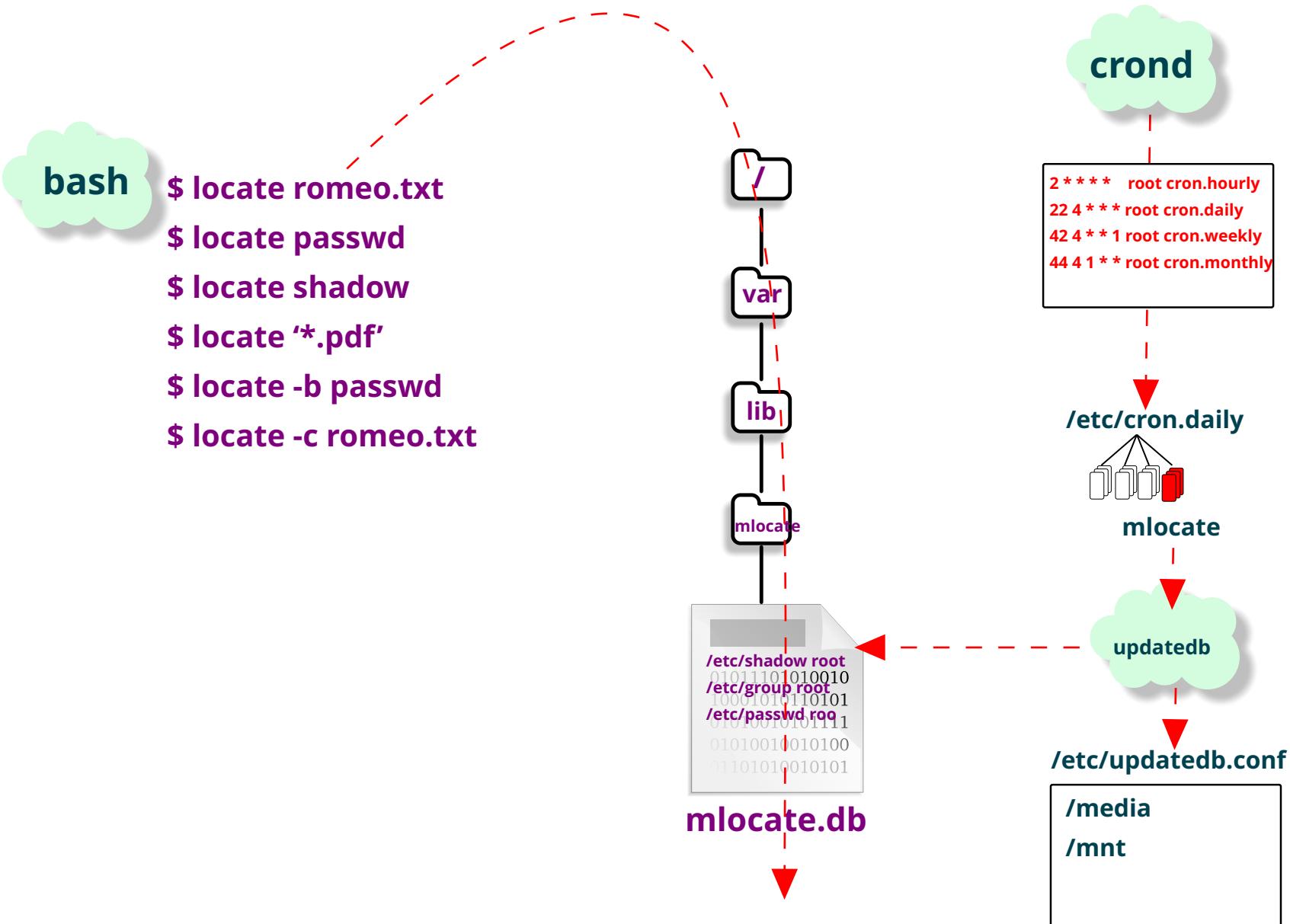


\$ cp fileA fileB
\$ type for
\$ type kill
\$ type vim
\$ whereis vim
\$ which vim

\$ gnome-<tab> -----> # command completion
\$ gnome-<tab><tab> # display all matching commands
\$ ls -l av-<tab> -----> # filename completion
\$ ls -l av-<tab><tab> # display all matching file and directories

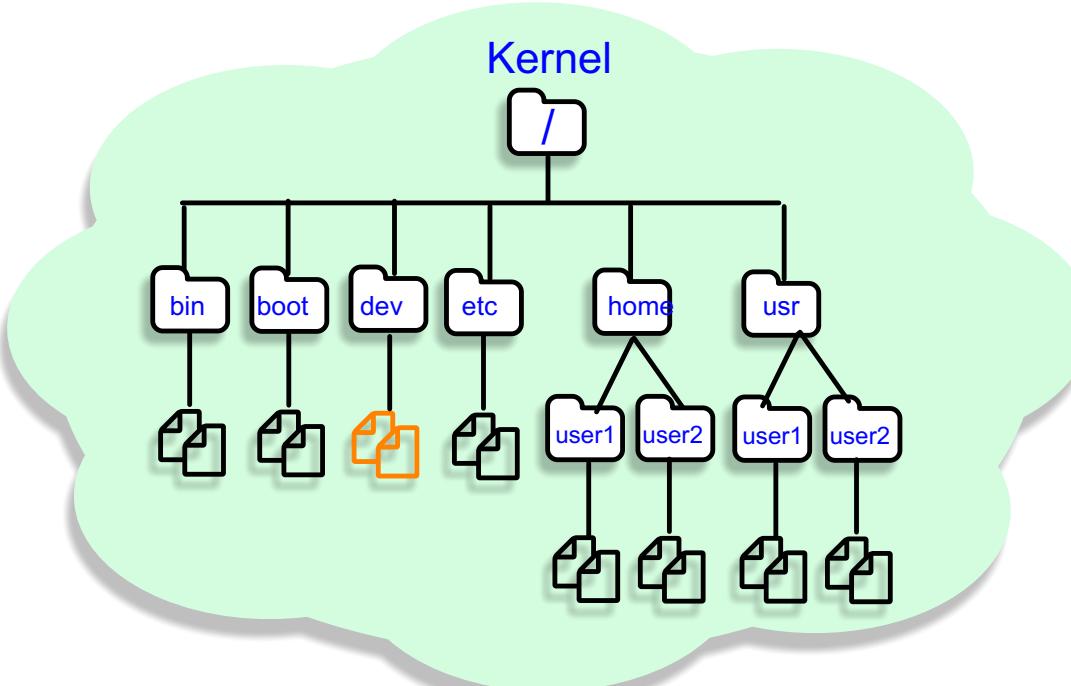


Searching the file system using locate



Searching the File System using 'find'

```
$ ls -li romeo.txt → 123032 -rwxr-xr--. 4 user10 QALXESS 453 Jul 14 10:52 romeo.txt
```



Filters

- type f|d||c|b|s|p
- perm 644
- perm -644
- perm ug=rw
- perm ug+rw
- perm -u=x
- links 4
- inum 123032
- path './test' -prune

- user scott|501
- group dba|5000
- size 500c|k|M|G
- size -300c|k|M|G
- size +1000c|k|M|G
- mtime 5
- mtime +7
- mtime -1
- mtime -1 -daystart
- name 'core'
- name '*.pdf'
- newer 'master.doc'
- atime +-n
- ctime +-n
- mmin +-n
- amin +- n
- cmin +-n

<ul style="list-style-type: none"> -print -ls -delete -exec command {} \; -ok command {} \; 	<ul style="list-style-type: none"> # display path # display long listing # deletes item # execute command # interactive
--	--

Functions

Note: Some options might not be available on UNIX.

\$ man find # display other filters and functions

Archiving using the dd (data definition) command

IBM Mainframe



EBCDIC
BigEndian
MSB

10101100011111010

```
dd if=unix_file of=ibm_file conv=swab,ebcdic
```

UNIX/Linux platform

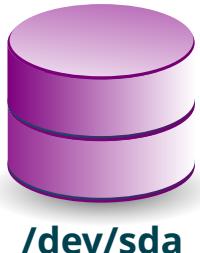


ASCII
LittleEndian
LSB

0101111000110101

```
dd < ibm_file > unix_file conv=swab,ascii
```

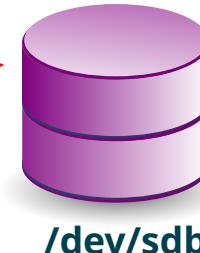
UNIX/Linux platform



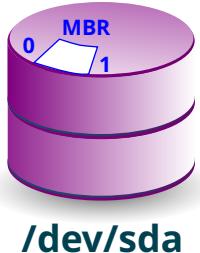
Create copy of disk/mirror

```
dd if=/dev/sda of=/dev/sdb  
dd if=/dev/sda2 of=/dev/sdb2
```

UNIX/Linux platform



UNIX/Linux platform



Create a backup of the MBR

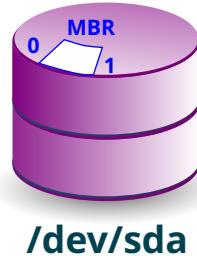
```
dd if=/dev/sda of=/var/tmp/MBR.bak bs=512 count=1  
dd if=/var/tmp/MBR.bak of=/dev/sda bs=512 count=1
```



File with MBR archive

Archiving using the dd (data definition) command

UNIX/Linux platform

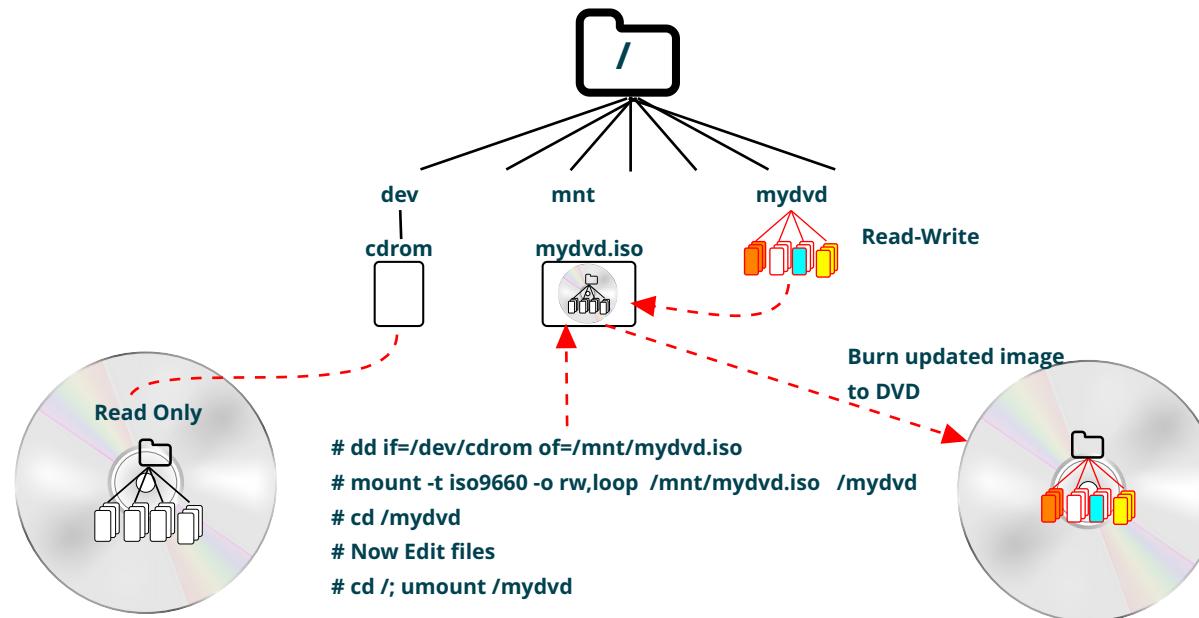


Create a swapfile for use with VM/paging

```
# dd if=/dev/zero of=/var/tmp/swapfile bs=10M count=5  
# chmod 600 /var/tmp/swapfile  
# mkswap /var/tmp/swapfile  
# swapon /var/tmp/swapfile  
# swapon -s  
# free
```



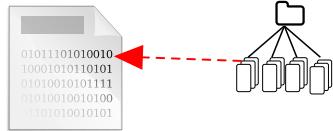
Rebuild a new DVD image with changes



Archiving using tar - Tape Archiving

Create: `tar cvf /dev/rst0 ./user5`

`tar cvf /var/tmp/home.tar ./user5`



Tabulate: `tar tvf /var/tmp/home.tar`

v=verbose (long listing)



Extract: `tar xvf /var/tmp/home.tar ./user5/romeo.txt`



Create: `tar zcvf /var/tmp/home.tar.gz ./user5`

Tabulate: `tar ztvf /var/tmp/home.tar.gz`

Extract: `tar zxvf /var/tmp/home.tar.gz ./user5/romeo.txt`

Append: `tar rvf /var/tmp/home.tar ./user5/new_dir`

Update: `tar uvf /var/tmp/home.tar ./user5/modified_dir`

Compress/Uncompress

`compress -v home.tar`

`uncompress home.tar.Z`



`gzip home.tar`

`gunzip home.tar.gz`



`bzip2 home.tar`

`bunzip2 home.tar.bz2`



`xz home.tar`

`unxz home.tar.xz`



Z=compress/uncompress

z=gzip/gunzip

j=bzip2/bunzip2

J=xz/unxz

Better
Compression
Slower

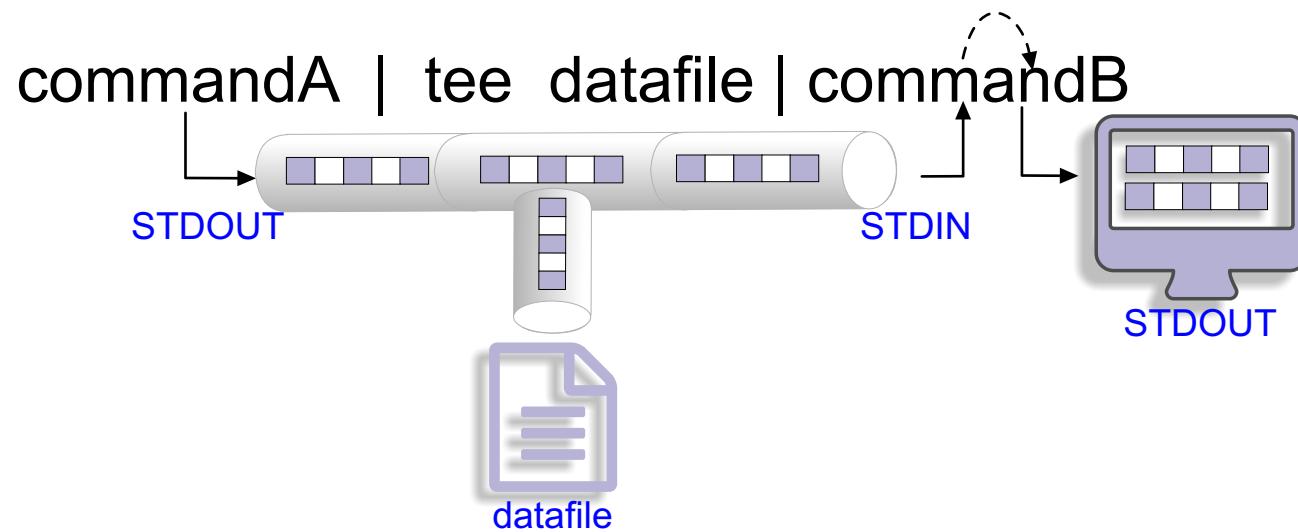
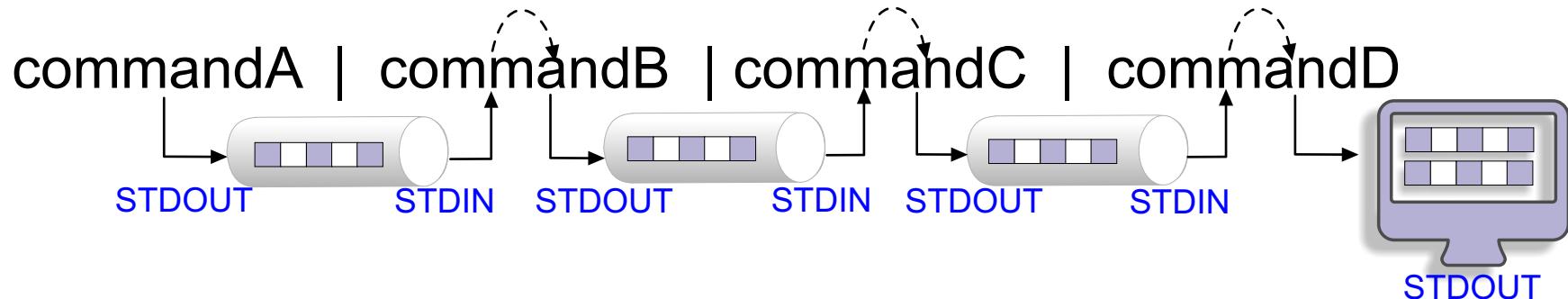
Alternative to tar/compress:

`$ zip --symlinks --quiet -r home.zip ./user5`

`$ unzip home.zip`

`$ find / -name '*.gif' | zip images -@`

Shell Pipes/Filters



Examples

```
$ ls -l /etc | wc -l  
$ ls -l /etc | more  
$ ps -ef | less  
$ ps -ef | head  
$ ps -ef | tail -n20
```

```
$ tail -n+100 words | head -n50  
$ ps -ef | grep '[a]td'  
$ ps -ef | grep 'firefox'  
$ who | wc -l  
$ ls -l | tr -s '\011\040' | cut -d ' ' -f1,5,9 | tr '\040' '\011'
```



Common commands used in Shell Pipes

```
$ cut -c1 /etc/passwd  
$ cut -c1,5,10 /etc/passwd  
$ cut -c1-10,40- /etc/passwd
```

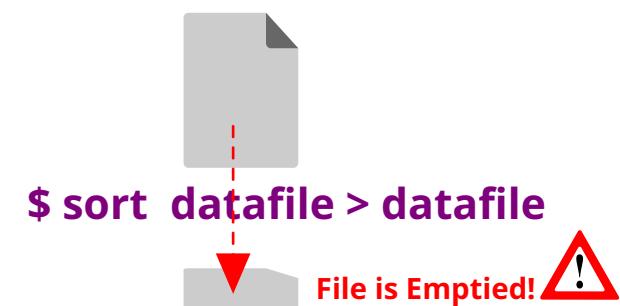
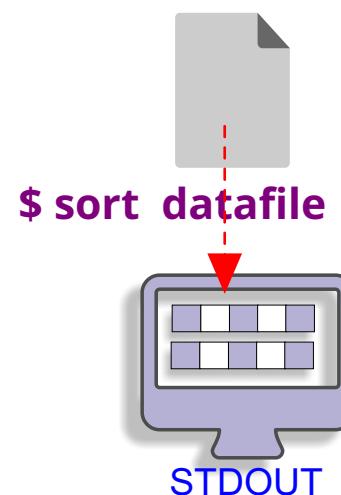
```
$ cut -d':' -f1 /etc/passwd  
$ cut -d':' -f1,3,7 /etc/passwd  
$ cut -d':' -f1-3 /etc/passwd
```

```
$ ls -l | cut -c1  
$ ls -l | cut -c2-10,49-  
$ ls -l | cut -d' ' -f1,5,9
```

```
$ tr abc ABC < /etc/passwd
```

```
$ tr 'a-z' 'A-Z' < /etc/passwd
```

```
$ ls -l | tr 'a-zA-Z' b-zA-Z  
$ ls -l | tr '' ':'  
$ ls -l | tr -s '\011\040' | cut -d' ' -f1,5,9  
$ ls -l | tr -d '0-9' > newdoc
```



```
$ sort -t':' -k3n,3 /etc/passwd # Numerical sort  
$ sort -t':' -k3nr,3 /etc/passwd # Reverse sort  
$ sort -t':' -k5,5 -f /etc/passwd # Ignore/Fold case
```

```
$ ls -l | sort  
$ ls -l | sort -k5nr,5 | head  
$ ls -l | sort -k6M,6 # Month order  
$ ls -l | sort -k6M,6 -k7n,7 -k8n,8
```

```
$ sort longlist | uniq # Removes duplicates  
$ sort longlist | uniq -c # count duplicates  
$ sort longlist | uniq -d # display duplicates  
$ sort longlist | uniq -u # exclusively unique
```



Common commands used in Shell Pipes

```
$ grep 'string' fileA fileB fileC *      # print lines which have strings
```



```
$ grep 'root' /etc/passwd  
$ grep 'user10' /etc/passwd  
# grep 'ssh' /var/log/messages  
# grep 'sudo' /var/log/secure
```

grep options

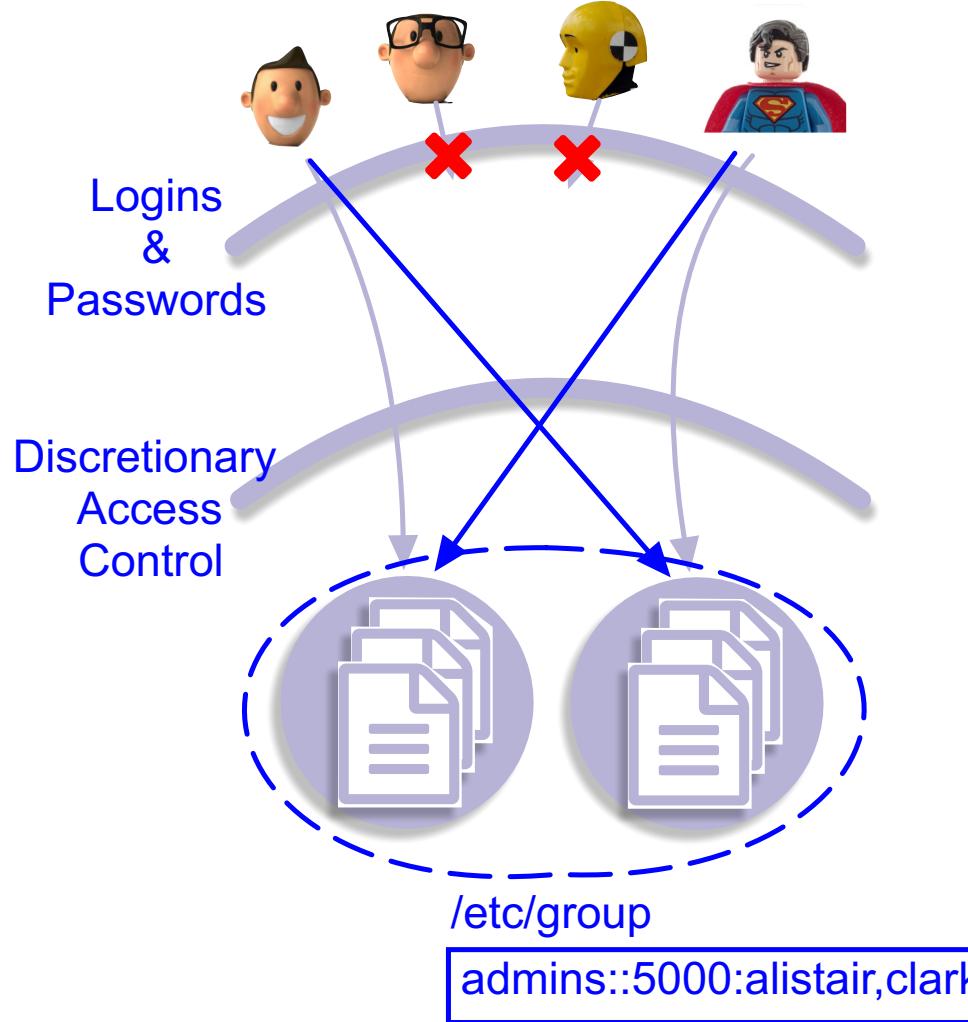
\$ grep -c 'streaming' /usr/share/dict/words	# count lines
\$ grep -v 'streaming' /usr/share/dict/words	# inverse search
\$ grep -i 'streaming' /usr/share/dict/words	# ignore cAsE
\$ grep -n 'streaming' /usr/share/dict/words	# display line numbers
\$ grep -l 'streaming' /usr/share/dict/words	# list filename if matched
\$ grep -B3 -A3 'streaming' /usr/share/dict/words	# display 3 lines Before and After match

```
$ ps -ef | grep '[s]shd' | tr -s '\011\040' | cut -d' ' -f1 # return PID field for sshd  
# kill -9 $(ps -ef | grep '[s]shd' | tr -s '\011\040' | cut -d' ' -f1)
```

```
$ ps -ef | awk '/[s]shd/{print $2}'  
# kill -9 $(ps -ef | awk '/[s]shd/{print $2}')
```



File and Directory Permissions



Actions



- r - Read (View)
- w - Write (Modify)
- x - Execute (Run)



- r - Read (List)
- w - Write (Create/Delete)
- x - Execute (Search)

File and Directory Permissions



9 Permissions

user	group	other
rwx	rwx	rwx

user	group	other
4 2 1	4 2 1	4 2 1

symbols	Binary	Octal
---	000	0
--x	001	1
-w-	010	2
-wx	011	3
r--	100	4
r-x	101	5
rw-	110	6
rwx	111	7

Changing Permissions on Existing Files/Dir

1. Octal Method (all 9 permissions)

```
$ chmod 750 f1 f2 f3 d1 d2 d3
$ chmod -R 750 f1 f2 f3 d1 d2 d3
```

2. Symbolic Method (fine tune)

```
$ chmod ugo+rwx f1 f2 f3 d1 d2 d3
$ chmod ug+rw f1 f2 f3 d1 d2 d3
$ chmod -R g-rwx f1 f2 f3 d1 d2 d3
$ chmod u+r,g-x,o-rwx f1 f2 f3
```

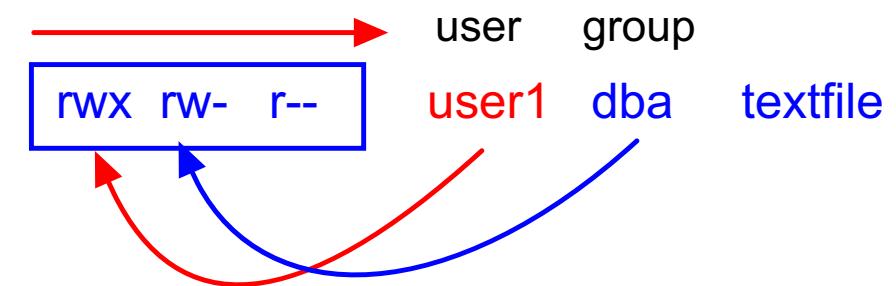
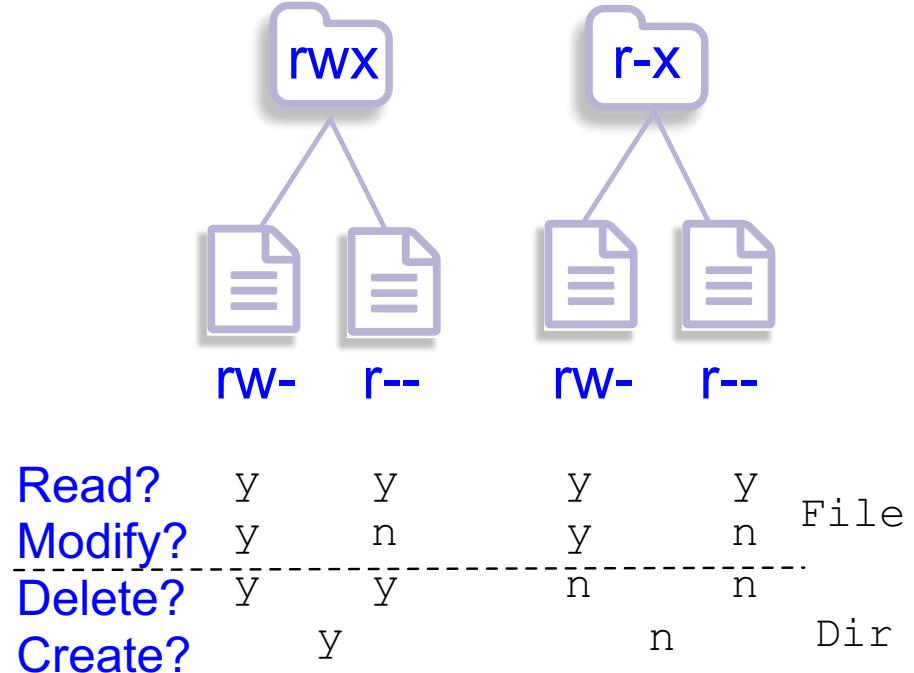


File and Directory Permissions

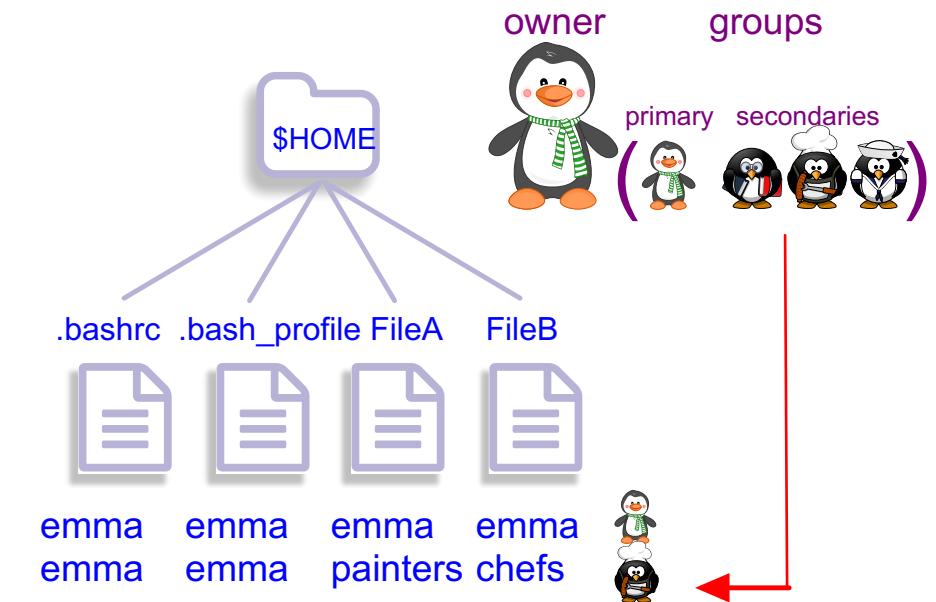
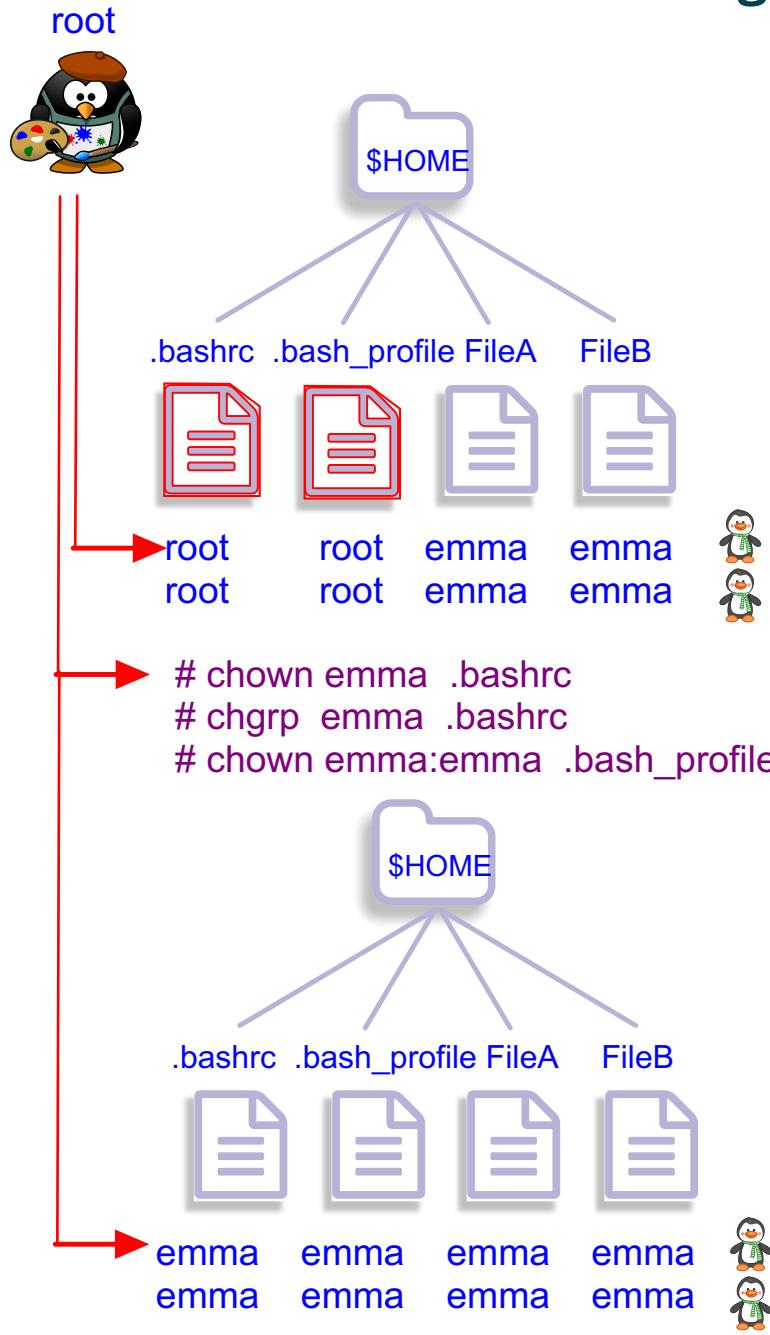
Permissions on NEW Files/Dir

FS default	666 rw-rw-rw-	777 rwxrwxrwx
Shell umask	022 ----w--w-	022 ----w--w-
Actual Permissions	644 rw-r--r--	755 rwxr-xr-x

\$ umask
\$ umask 022
\$ umask 027
\$ umask 077



Changing Owner and Group



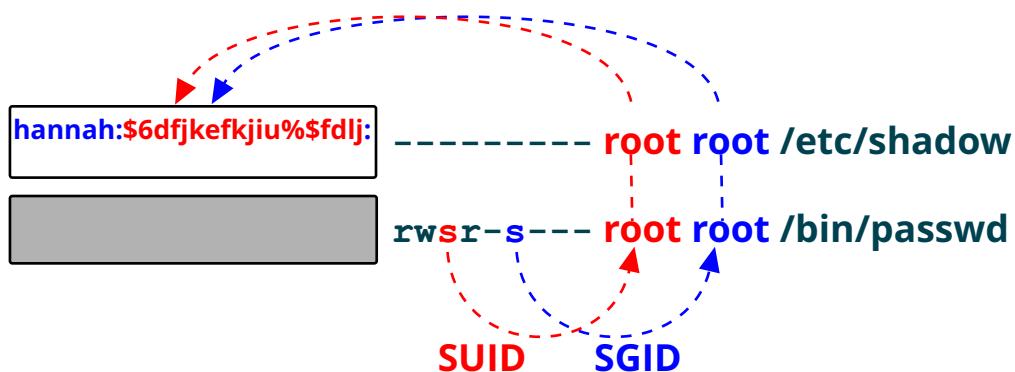
```
$ groups
emma painters chefs sailors
$ chgrp painters FileA
$ chgrp chefs FileB
$ chgrp -R chefs $HOME
```

Only root can use chown

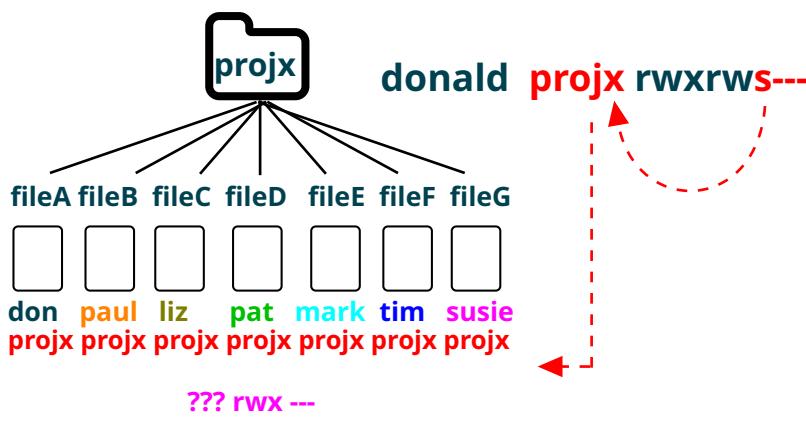
Owners can use chgrp

Advanced Permissions and attributes

SUID and GID on a file (program)

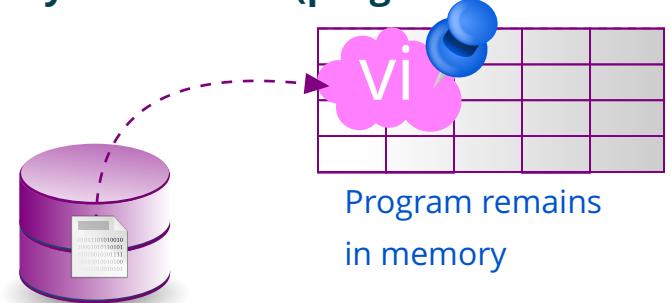


SGID on a directory

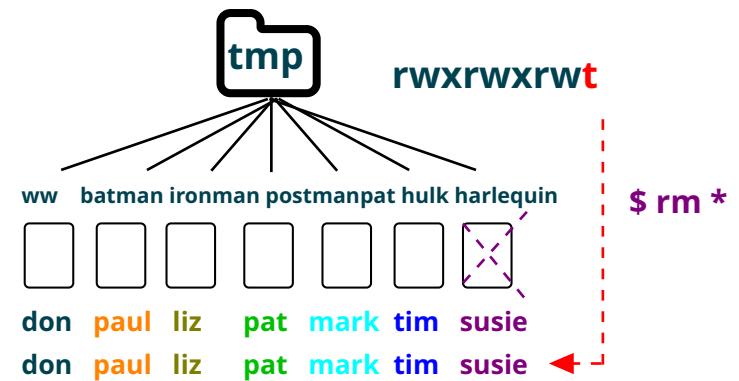


	SUID	SGID	Sticky Bit
File	x -	Run as owner s s	Run as group owner s t
Dir	x -	s Inherit dir group name s	t Only owner can delete t

Sticky bit on a file (program) = obsolete



Sticky bit on a directory

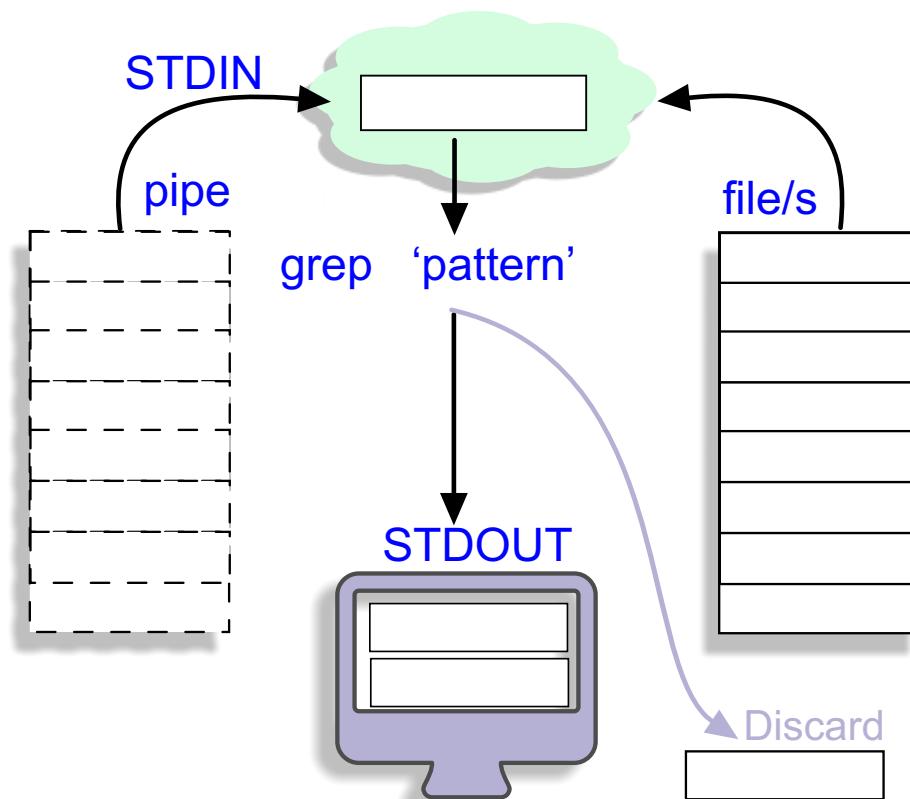


USER GROUP OTHER

# chmod 4750 /etc/shadow	or chmod u+s
# chmod 2770 /project	or chmod g+s
# chmod 1777 /tmp	or chmod o+t

```
$ find / -perm -4000 -ls 2> /dev/null
$ find / -perm -2000 -ls 2> /dev/null
$ find / -perm -1000 -ls 2> /dev/null
```

Searching/Filtering Files and Data Streams



Examples

grep '^root' /etc/passwd
ps -ef | grep 'atd'

R.E/RegExp/Regular Expression Patterns

Line Anchors

Start of line	'^the'
End of line	'ing\$'
Blank line	'^\$'

Single Char Class

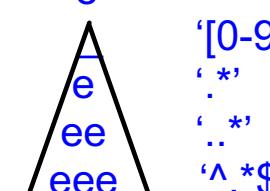
Match 1 x any char	'.'
Match 1 x list	'[abcd]'
Match 1 x range	'[a-zA-Z]'
Match 1 x not range	'[^aeiou]'

Escape Char

Match literal char	'\.'
--------------------	------

Repetition Char Class

Match 0 or more of preceding char	'e*' '[0-9]*'
	'[0-9][0-9]*'
	'*' '..'
	'*' '^.*\$'
	'.' '.'



Word Anchor Char Class

Start of Word	'<million'
End of Word	'million>'
Word	'<million>'



RE, Regex, Regular Expressions: B.R.E/E.R.E

Line Anchors

Start Line '`^the`'

End Line '`ing$`'

Blank lines '`^\$`'

Single char class

'.' matches 1 x any char

'`.onald`'

'`^.....$`'

Limited range char class

'`[abc]`' matches 1 x list

'`[A-Z]`' matches 1 x range

'`[a-zA-Z]`' matches 1 x ranges

'`[^0-9]`' matches 1 x **not** range

'`[dD]onald`'

'`^[A-Z]`'

'`[0-9]`'

'`[aeiou][aeiou][aeiou]`'

Escape char class

'\.' escape next 1 x char

Repetition char class

'`e*`' repeat prev char **0 or more**

'`e`' '`[0-9]*`'

'`ee`' '`[0-9][0-9]*`'

'`eee`' '`.*`'

'`eeee`' '`..*`'

'`^.*$`' match entire line

Example from vi/vim

:1,10s/`^.*$`/ &/g

Interval Repetitions/Quantifiers

'`:[0-9]\{10\}:`' Exact repetitions

'`:[0-9]\{10,20\}:`' Min,max repetitions

'`:[0-9]\{10,\}:`' At least repetitions



: : :_{15 digits}:

'`:[-]\{0,1\}[0-9]\{15\}:`'

Word Anchors

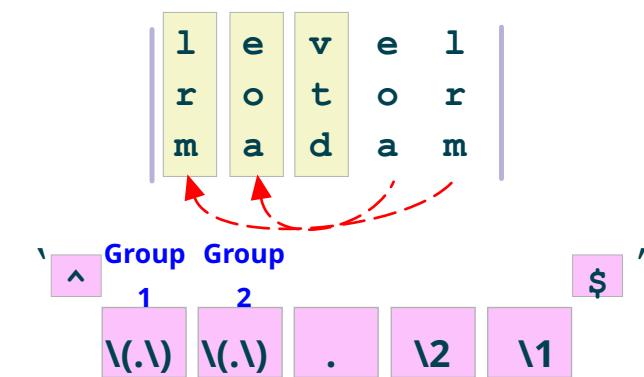
Start word '`\<million`'

End word '`million\>`'

Word '`\<million\>`'

(grep -w 'million')

Grouping/Expression/Back referrals



Extended Regex/E.R.E

Repetitions

'`e?`' repeat **0 or once**

'`e+`' repeats **1 or more**

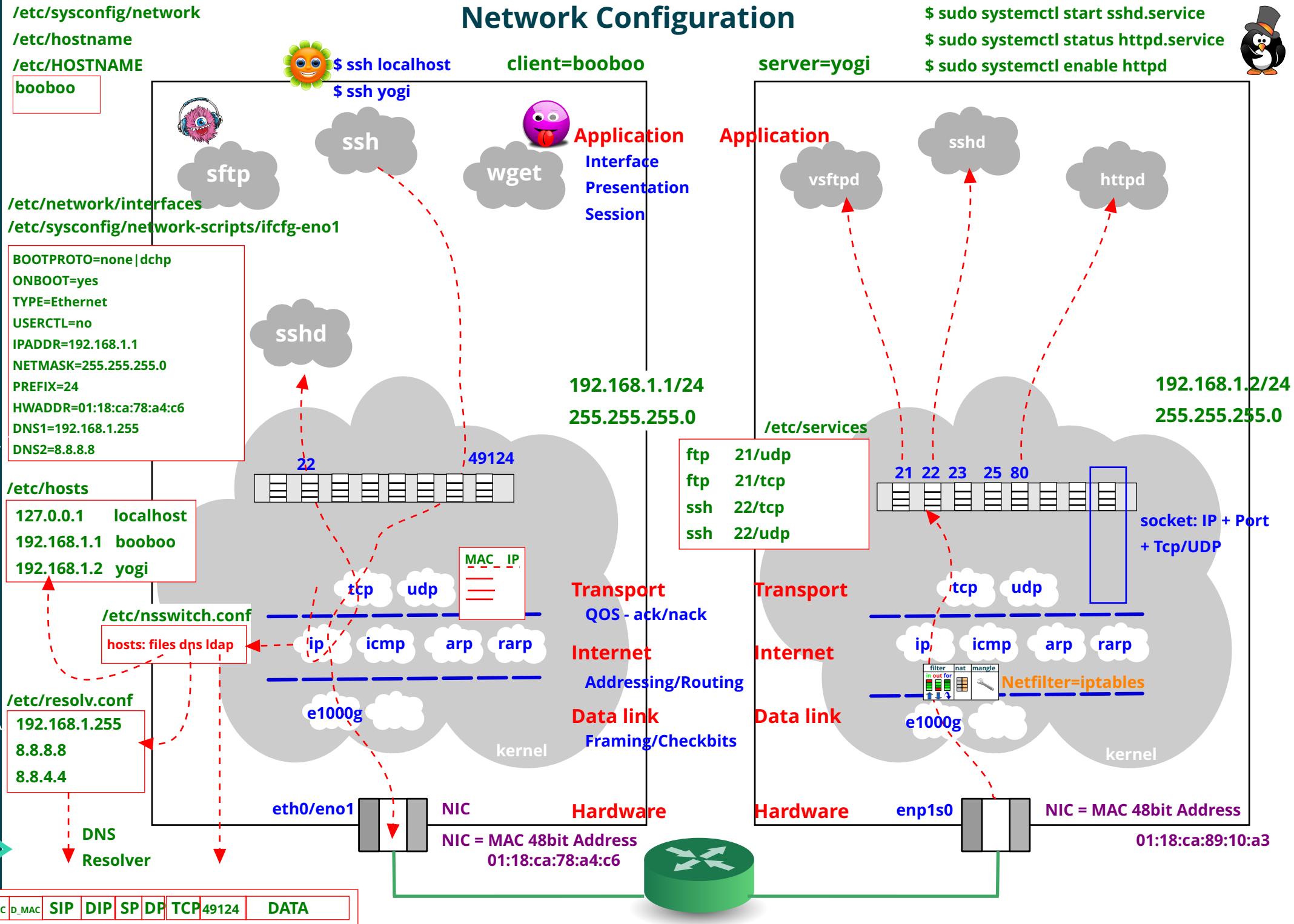
Alternation (OR)

'`donald|dcameron|Sir Cameron`'

Grouping

'`(rhubarb)+`' grep -E '`:[-]?[0-9]\{15\}:`'

Network Configuration



Accessing remote machines

`ftp yogi`

`telnet yogi`

`rcp -r $HOME yogi:/tmp`

`rlogin yogi`

`rsync -av -HAX photos yogi:/tmp`

`ssh yogi`

`ssh yogi -l root`

`ssh root@yogi`

login & password
(clear ascii text)

20/21

23

513

514

22

ftpd

vsftpd

telnet

rcpd

rlogind

sshd

/etc/passwd

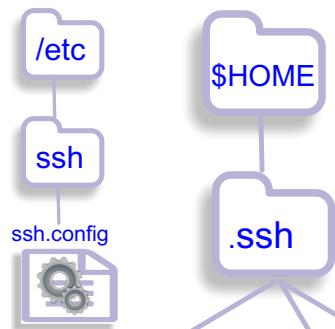
/etc/shadow



Host Authentication
User Authentication
Encryption

\$ ssh-keygen

Enter passphrase
Re-enter passphrase



\$ ssh-copy-id yogi

id_rsa.pub

