Linux Architecture, Features & Vi Editor

BENNETT UNIVERSITY
THE TIMES GROUP

Dr. Vimal Kr Baghel (Course Instructor), Assistant Professor
School of Computer Science Engineering & Technology (SCSET)
Bennett University Greater Noida

BENNETT UNIVERSITY
THE TIMES GROUP

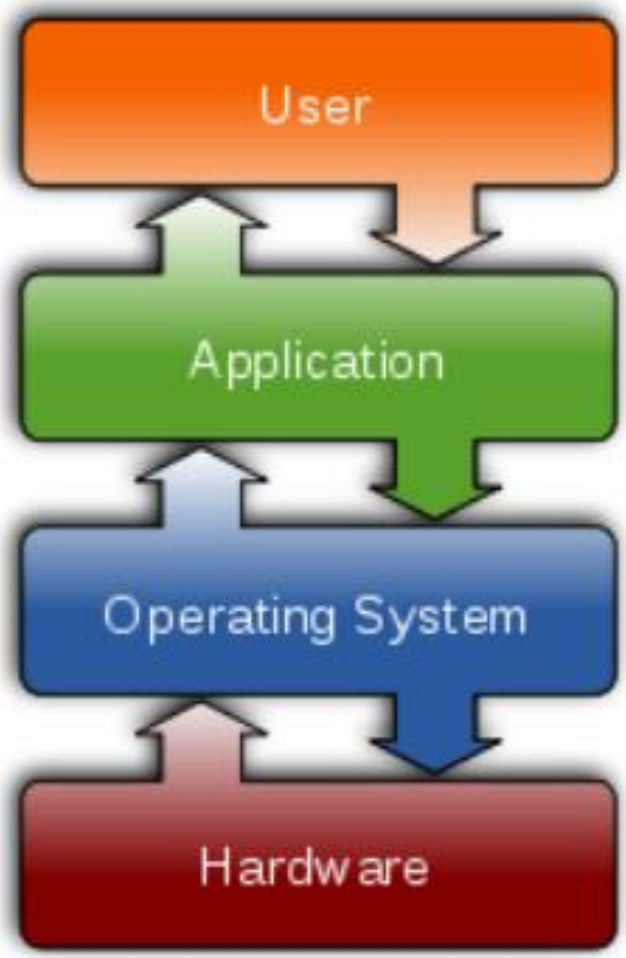
Outline

- Operating System
- Linux features and components
- Linux Architecture
- Shell
- Vi Editor
- Q & A

BENNETT UNIVERSITY
THE TIMES GROUP

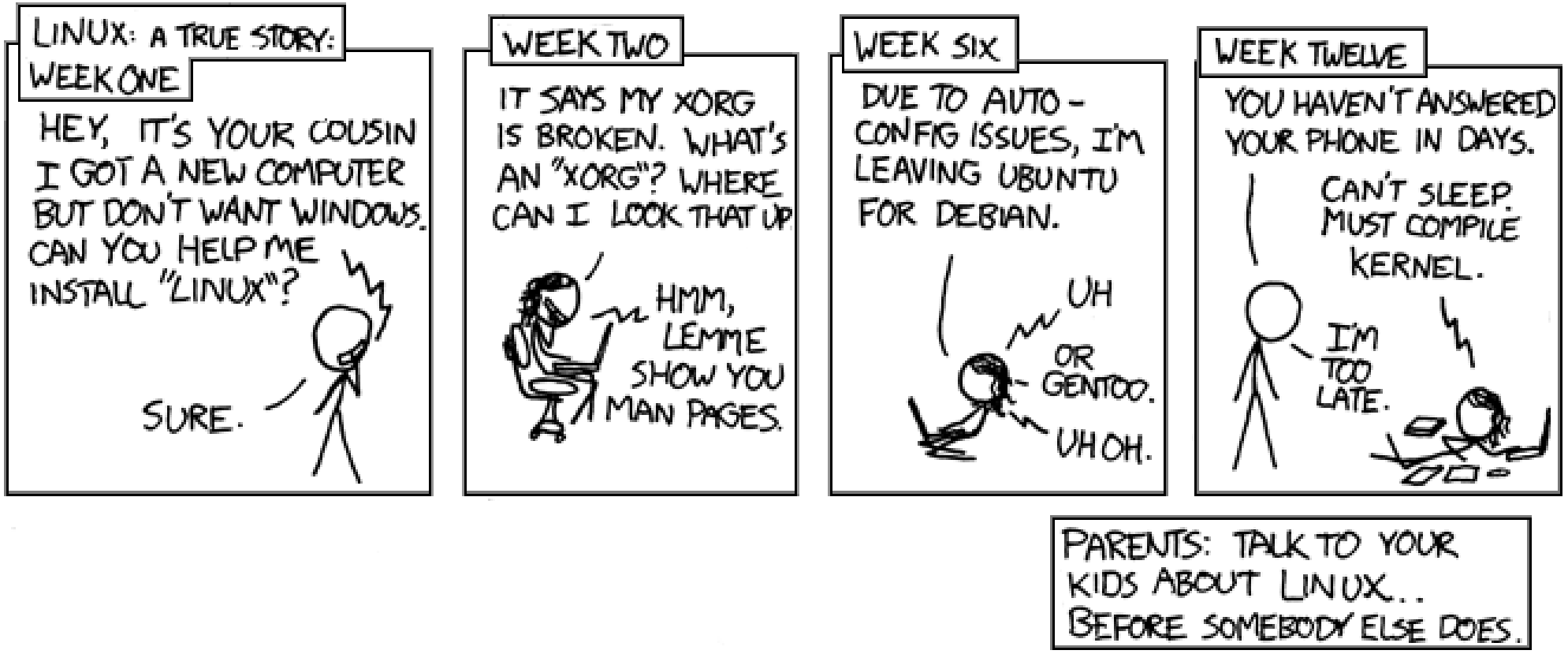
Operating System (OS)

- What is an OS?
- Why OS?
- How it works?
- Is there only one OS in a machine?
- Types of OS?



On to Linux

- Courtesy XKCD.com



Linux

BENNETT UNIVERSITY
THE TIMES GROUP

Linux: A kernel for a Unix-like operating system.

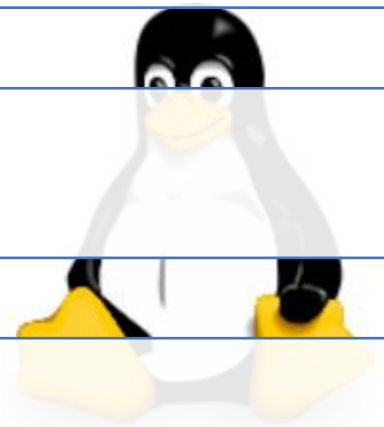
- commonly seen/used today in servers, mobile/embedded devices, ...

GNU: A "free software" implementation of many Unix-like tools

- many GNU tools are distributed with the Linux kernel

distribution: A pre-packaged set of Linux software.

- examples: Ubuntu, Fedora



key features of Linux:

- **open-source software**: source can be downloaded
- free to use
- constantly being improved/updated by the community

BENNETT UNIVERSITY
THE TIMES GROUP

Linux Operating System

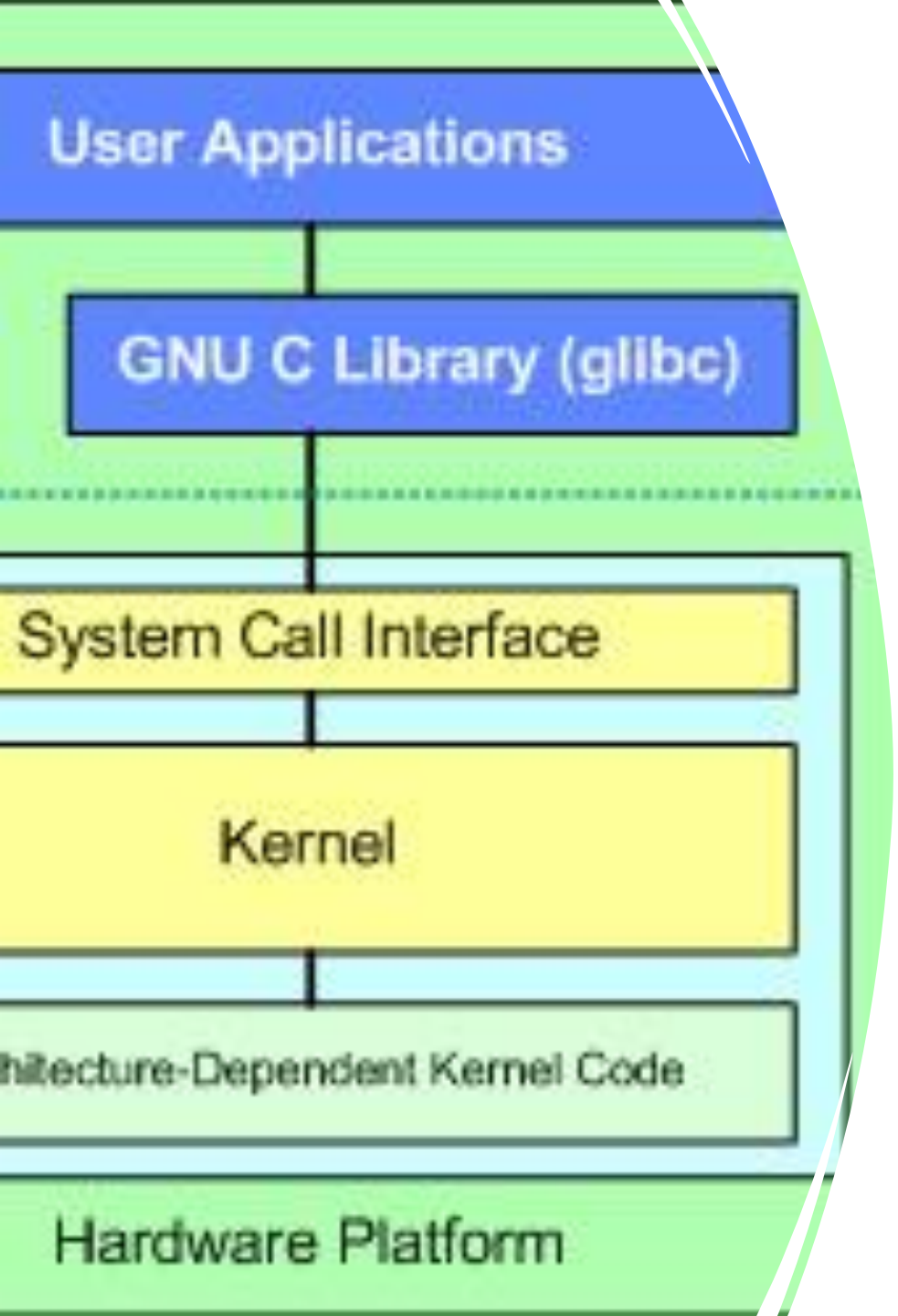
- Minix, the first open-source operating system, written by Andrew S. Tanenbaum in C, about 12000 lines of code.
- 1991, first Linux kernel written in C by **Linus Torvalds**, University of Helsinki, Finland.
- It was developed with the contribution of many programmers around the world.
- It is functionally like Unix (a clone).
- 1993 – FreeBSD 1.0 (Berkley Unix), 1994 – RedHat Linux is introduced.
- 1999 – Linux available for PowerPC (Apple)
- Now – adopted by many companies and most universities, third world countries.
- Standard for parallel and high-performance computing (clusters).
- Available for most computers, including PDA, supports graphical user interfaces, networking, and has many applications.

Features of Linux?

- It's free!
  - the source code is also available, and anybody can write their own Linux if they include the source code in the distribution.
- Most users consider it a more stable and reliable OS than Windows.
- It's an alternative to Microsoft's dominance of the software market.
- It is multi-tasking, multi-user, and good support of multiple CPUs.
- Many utilities and APIs are now included in most distributions, like the g++ compiler, OpenGL, MPI, pthreads, etc.
- Mac OS now has an integrated shell and can run X11, Linux-specific applications.

Components of Linux

- The *kernel* – the core of the OS that controls the resources.
- A hierarchical file system (*FHS*)
- *Shells* – applications that interpret the commands from the user. They are active in the textual mode or terminal mode. Shells can also execute script files. Examples: bash, tcsh, zsh, sh, etc.
- *Graphical interfaces* – the X window system. Desktop interfaces: Gnome, KDE, fvwm, etc.
- *Specific libraries:* X11, gtk-glib-gnome, Qte, etc.

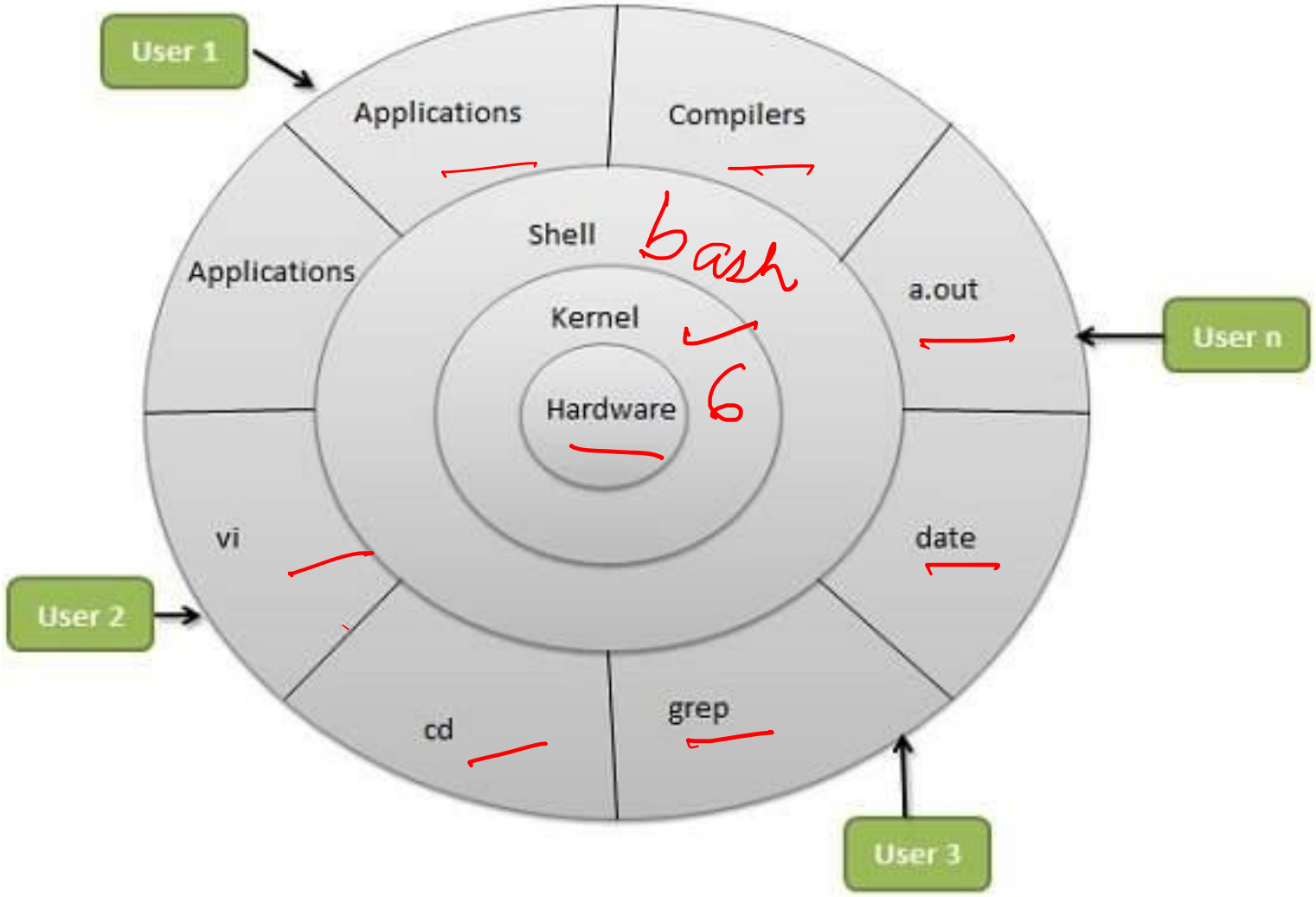


BENNETT UNIVERSITY
THE TIMES GROUP

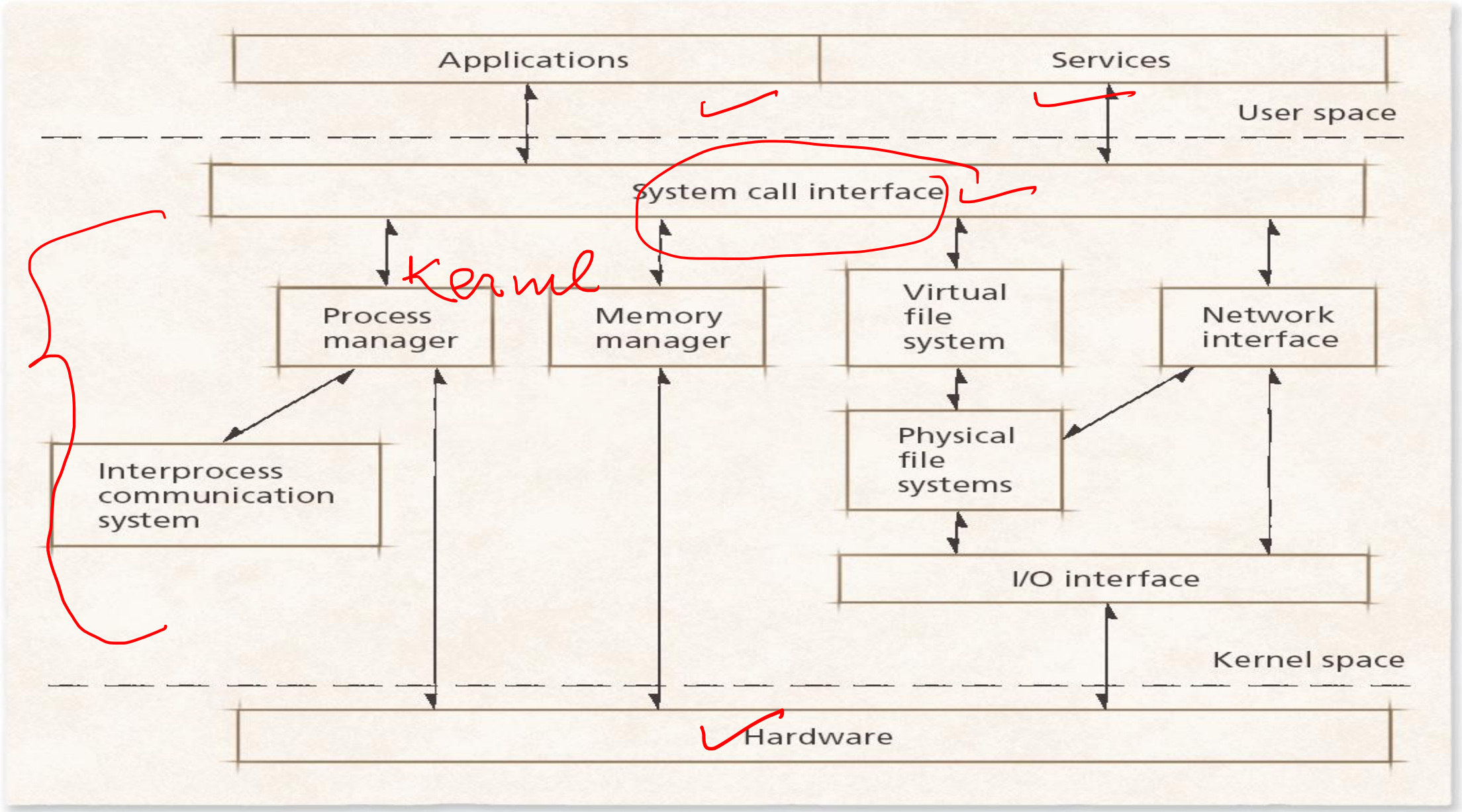
Linux Architecture

- Monolithic kernel
  - Contains modular components
- UNIX-based operating system
- Six primary subsystems:
  - Process management
  - Inter-process communication
  - Memory management
  - File system management
    - VFS: provides a single interface to multiple file systems
  - I/O management
  - Networking

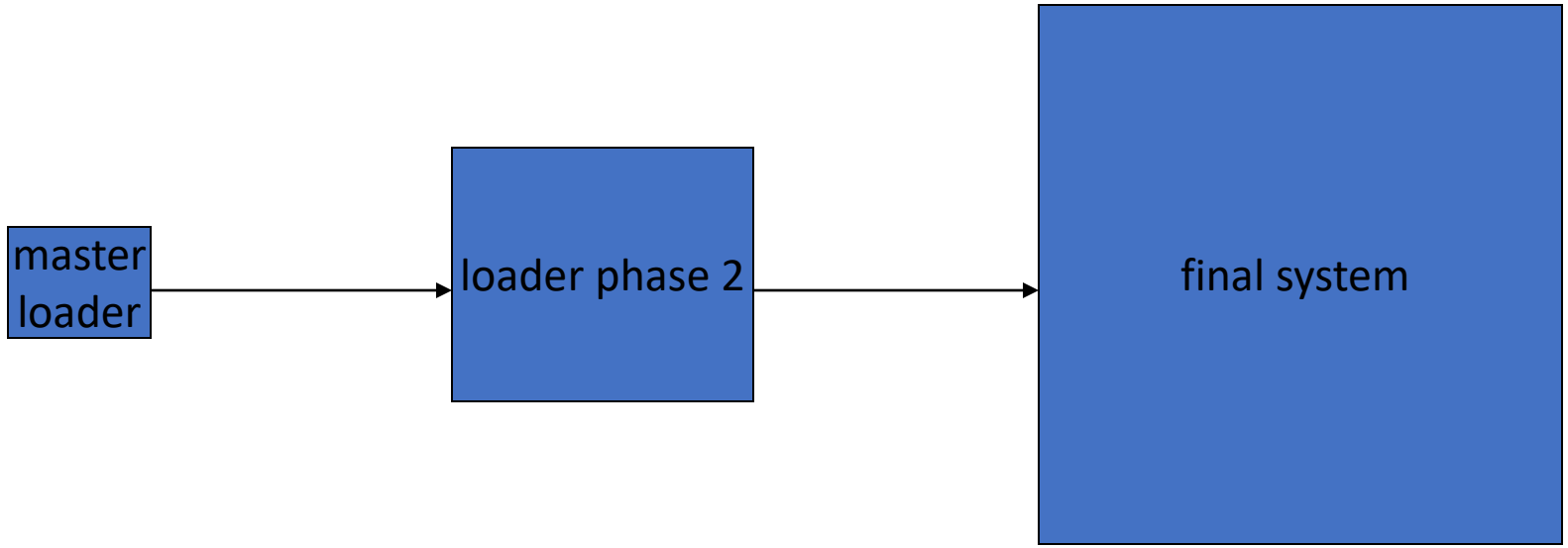
Linux Architecture



Linux Kernel Architecture

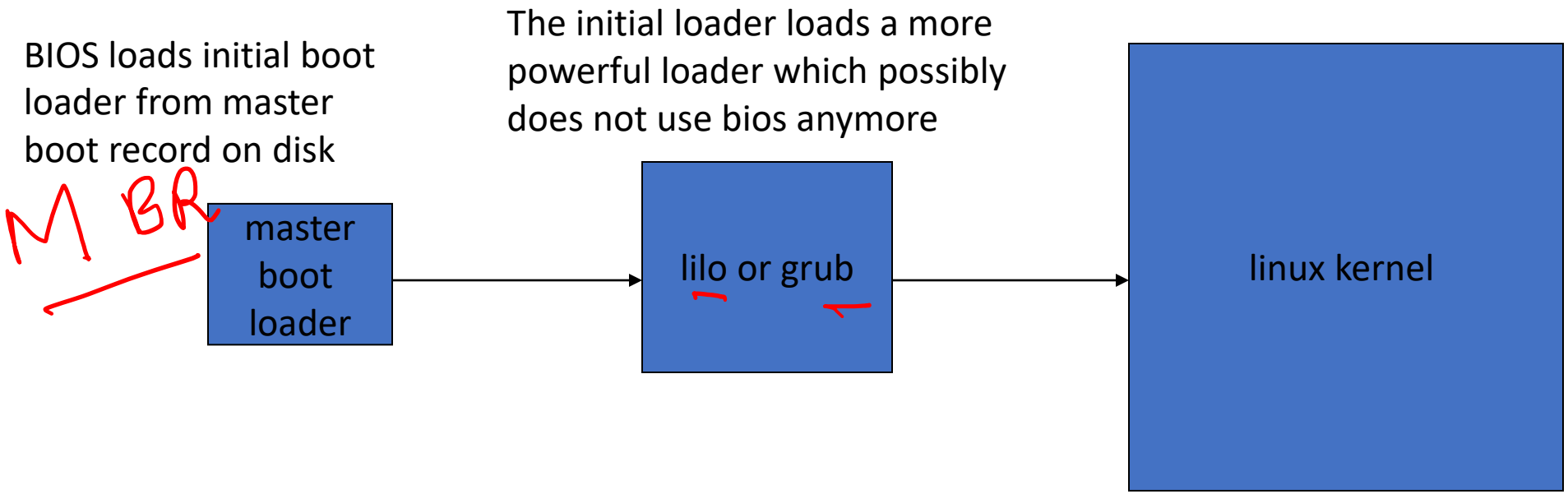


Bootstrapping



- bootstrapping is the process of starting a large and complicated system via a number of very small steps.
- A characteristic feature of bootstrapping is that the wonderful and powerful functions of a large system cannot be used to start the system itself – they are simply not yet available because the system is not running.

Loading Linux (1)



- Linux loaders are lilo or grub which are both found under /boot.
- The difference is that lilo knows exactly at which block of the partition the linux kernel starts and how big it is.
- It does NOT understand the linux filesystem and takes the information about the kernel from when the kernel was installed under /boot and lilo was re-run.
- Grub understands the filesystem and can locate the kernel within.

Loading Linux (2): System Check and Autoconfiguration

BENNETT UNIVERSITY
THE TIMES GROUP

- ✓ During system start the following functions are performed:
  - determine CPU type, RAM etc.
  - stop interrupts and configure memory management and kernel stack
  - Initialize rest of kernel (buffers, debug etc.)
  - Start autoconfiguration of devices from configuration files and via probing hardware addresses.

- ✓ Probing is done via device driver routines.
- ✓ It means that certain memory locations are checked for the presence of a device.
- ✓ The system catches errors and then assumes that there is nothing mapped to this location.
- ✓ During regular operation, such errors would cause a kernel panic.

Loading Linux (3): Start processes

- ✓ **init**: the first process and the only one started by the kernel itself. Starts other processes e.g. User is waiting for logins from terminals
- ✓ Swapper and other system processes (yes, the kernel depends on processes running in user mode)

✓ Init is the parent of all processes. Killing it usually causes an immediate shutdown of the whole system.

Loading Linux (4): Go to runlevel

- ✓ System configuration scripts under /etc/rc.d/ are executed (shell scripts)
- ✓ Depending on the configured **runlevel** the system either boots into single-user mode or multi-user mode with or without networking and with or without X Window system. (The runlevel can be specified at kernel load-time)

✓ Shell scripts basically initialize the whole system once the kernel itself is running.

Shell

Shell is the user interface to the operating system

Functionality:

- Manage files (wildcards, I/O redicrection)
- Manage processes (build pipelines, multitasking)

Most popular shells:

- The Bourne shell (sh)
- The Korn shell (ksh)
- The C shell (csh)
- The Bourne Again shell (bash)

Shell

- **The Bourne shell /bin/sh** (S. R. Bourne, 1977)
  - powerful syntactical language
  - strong in controlling input and output
  - expression matching facilities
  - ☹ interactive use: the use of shell functions.
- **The C-shell /bin/csh** (Bill Joy, UCB, 1978)
  - new concepts: job control and aliasing
  - much better for interactive use
  - ☹ different input language:
    - out went the good control of input and output
    - too buggy to produce robust shell scripts

Shell

/bin/tsch (Ken Greer, Carnegie Mellon University, late 1970s)

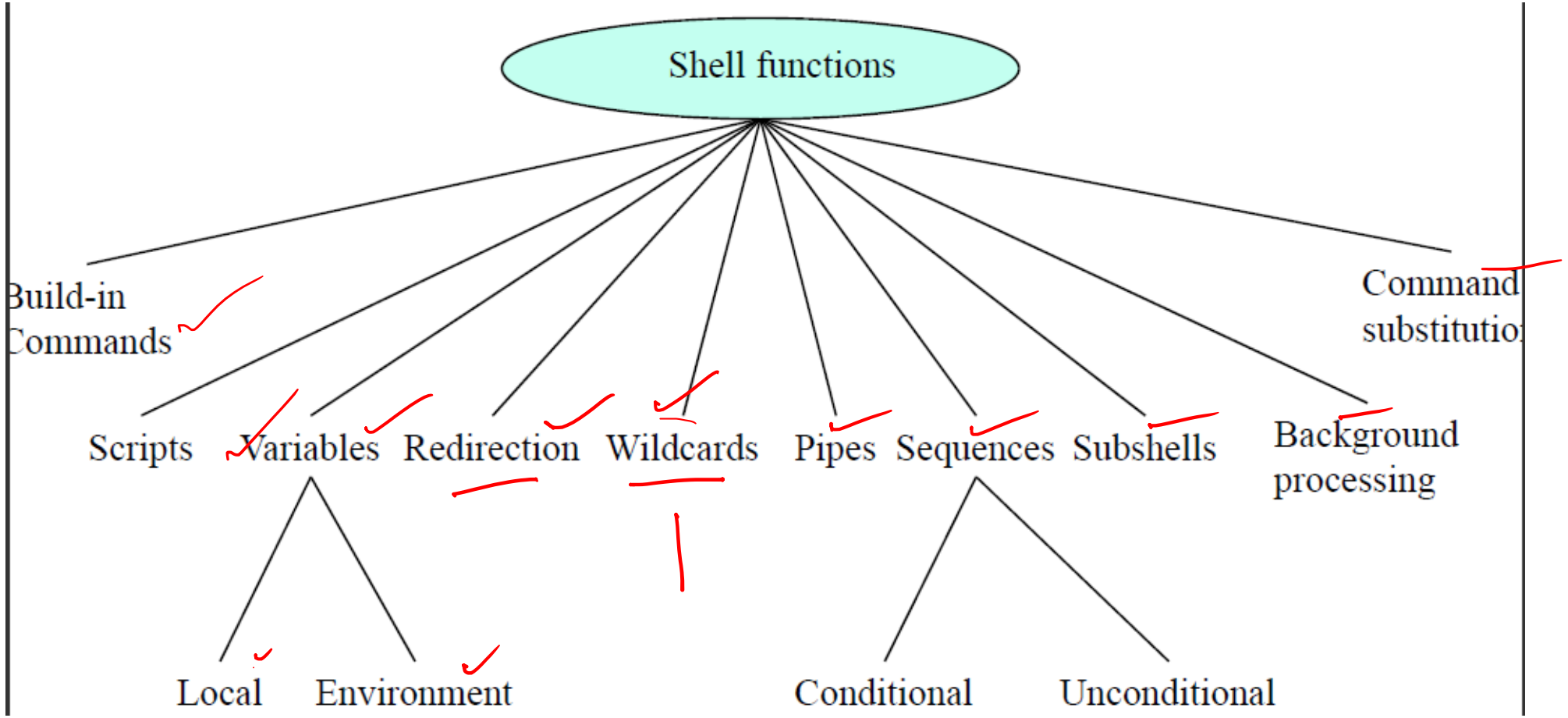
- User –oriented command line editing
- Out most of the bugs

Korn shell /bin/ksh (David Korn, AT&T, early 1980s)

- Bourne shell language
- C shell's features for interactive work
- You had to pay AT&T for it!

GNU project: a free shell=> **/bin/bash (the Bourne again shell)**

Core Shell Functionality



Selecting a Shell

```
[vimal@baghel ~]$ echo $SHELL
/bin/bash
[vimal@baghel ~]$ bash
[vimal@baghel ~]$ exit
exit
[vimal@baghel ~]$ ksh
$ exit
[vimal@baghel ~]$ sh
sh-3.2$ exit
exit
[vimal@baghel ~]$ csh
[vimal@baghel ~]$ exit
exit
```

Changing shell

- To change your default shell use the ***chsh*** *utility* which requires full pathname of the new shell

```
[c33235@snowball ~]$ chsh
Changing shell for c33235.
Password:
New shell [/bin/bash]: /bin/tcsh
Shell changed.
[c33235@snowball ~]$
```

What is vi ?

- The *visual editor* on the Unix.
- Before vi the primary editor used on Unix was the line editor
  - User was able to see/edit only one line of the text at a time
- The vi editor is not a text formatter (like MS Word, Word Perfect, etc.)
  - you cannot set margins
  - center headings
  - Etc…

Characteristics of vi

The vi editor is:

- a very powerful
- but at the same time it is cryptic
- It is hard to learn, specially for windows users

The best way to learn vi commands is to use them

So Practice…

Vim equals Vi

- The current iteration of **vi** for Linux is called **vim**
  - **Vi** I**m**proved
  - http://www.vim.org

Starting vi

Type **vi** <filename> at the shell prompt

After pressing enter the command prompt disappears and you see tilde(~) characters on all the lines

These tilde characters indicate that the line is blank

Vi modes

There are two modes in vi

- Command mode
- Input mode

When you start vi by default it is in command mode

You enter the input mode through various commands

You exit the input mode by pressing the Esc key to get back to the command mode

How to exit from vi

First go to command mode

press **Esc** There is no harm in pressing **Esc** even if you are in command mode. Your terminal will just beep and/or or flash if you press **Esc** in command mode

There are different ways to exit when you are in the command mode

How to exit from vi (comand mode)

:q <enter> is to exit, if you have not made any changes to the file

:q! <enter> is the forced quit, it will discard the changes and quit

:wq <enter> is for save and Exit

:x <enter> is same as above command

ZZ is for save and Exit (Note this command is uppercase)

The **!** Character forces over writes, etc. **:wq!**

Moving Around

- You can move around only when you are in the command mode
- Arrow keys usually works(but may not)
- The standard keys for moving cursor are:
  - **h** - for left
  - **l** - for right
  - **j** - for down
  - **k** - for up

Moving Around

w - to move one word forward

b - to move one word backward

$ - takes you to the end of line

<**enter**> takes the cursor the the beginning of next line

Moving Around

- **-** - (minus) moves the cursor to the first character in the current line
- **H** - takes the cursor to the beginning of the current screen(Home position)
- **L** - moves to the Lower last line
- **M** - moves to the middle line on the current screen

Moving Around

f - (find) is used to move cursor to a particular character on the current line

- For example, **fa** moves the cursor from the current position to next occurrence of '**a**'

F - finds in the reverse direction

BENNETT UNIVERSITY
THE TIMES GROUP

Moving Around

) - moves cursor to the next sentence

} - move the cursor to the beginning of next paragraph

(- moves the cursor backward to the beginning of the current sentence

{ - moves the cursor backward to the beginning of the current paragraph

% - moves the cursor to the matching parentheses

BENNETT UNIVERSITY
THE TIMES GROUP

Moving Around

Control-d scrolls the screen down (half screen)

Control-u scrolls the screen up (half screen)

Control-f scrolls the screen forward (full screen)

Control-b scrolls the screen backward (full screen).

Entering text

- To enter the text in vi you should first switch to input mode
  - To switch to input mode there are several different commands
  - **a** - Append mode places the insertion point after the current character
  - **i** - Insert mode places the insertion point before the current character

Entering text

I - places the insertion point at the beginning of current line

o - is for open mode and places the insertion point after the current line

O - places the insertion point before the current line

R - starts the replace(overwrite) mode

Editing text

x - deletes the current character

d - is the delete command but pressing only d will not delete anything you need to press a second key

- **dw** - deletes to end of word
- **dd** - deletes the current line
- **d0** - deletes to beginning of line

There are many more keys to be used with delete command

The change command

c - this command deletes the text specified and changes the vi to input mode. Once finished typing you should press <**Esc**> to go back to command mode

cw - Change to end of word

cc - Change the current line

There are many more options

Structure of vi command

The vi commands can be used followed by a number such as

- **n<command key(s)>**
- For example **dd** deletes a line **5dd** will delete five lines.

This applies to almost all vi commands

This how you can accidentally insert a number of characters into your document

Undo and repeat command

u - undo the changes made by editing commands

. (dot or period) repeats the last edit command

BENNETT UNIVERSITY
THE TIMES GROUP

Copy, cut and paste in vi

yy - (yank) copy current line to buffer

nyy - Where **n** is number of lines

p - Paste the yanked lines from buffer to the line below

P - Paste the yanked lines from buffer to the line above

(the paste commands will also work after the **dd** or **ndd** command)

Creating a shell script using vi

Create a directory call **class**

Change into **class**

vi myscript.sh

inside the file enter following commands

- clear
- echo "==========="
- echo "Hello World"
- echo "==========="
- sleep 3
- clear
- echo Host is $HOSTNAME
- echo User is $USER

Creating a shell script using vi

- Save the file
- Change the permissions on myscript.sh

  chmod 700 myscript.sh <enter>
- Now execute myscript.sh

  myscript.sh <enter>
- Did the script run?
- Why not?
  - Hint, think about absolute vs relative path
  - Type **echo $PATH** to see your PATH variable
  - Try this **./myscript.sh** <enter>
  - The **./** mean right here in this directory!

BENNETT UNIVERSITY
THE TIMES GROUP

Thanks

Q & A