All things Linux

Nickalas Reynolds

Lecture Series: Day 1
The University of Oklahoma

October 29, 2018

email: casit@ou.edu

Table of Contents

What is Operating System

What is an OS?

1. Just a collection of programs (binary files) that talk to hardware/software
2. In charge of process, memory, data, and I/O management
3. At first it was UNIX -> DOS -> MAC
4. All proprietary :(

What is an OS?

1. Just a collection of programs (binary files) that talk to hardware/software
2. In charge of process, memory, data, and I/O management
3. At first it was UNIX -> DOS -> MAC
4. All proprietary :(
5. Then Linus saved the day
6. 1991 (As a grad student in his free time) built the GNU/Linux Kernel (not an OS)
7. Now we have GNU/linux flavours (RHEL, Arch, Gentoo, Ubuntu, CentOS, Puppy, etc)

Architecture

Overall: Hardware) Kernel) Shell) users/programs
Communication: hardware $\rightleftharpoons$ kernel $\rightleftharpoons$ shells $\rightleftharpoons$ users

Architecture

Overall: Hardware) Kernel) Shell) users/programs
Communication: hardware $\rightleftharpoons$ kernel $\rightleftharpoons$ shells $\rightleftharpoons$ users
Remember.... EVERYTHING IS A FILE !!

*/	$\leftarrow$	root : sudo access only !!
/bin	$\leftarrow$	root : system Binary files (programs)
/proc	$\leftarrow$	root : process + hardware info
/etc	$\leftarrow$	root : system config files
/lib(64)	$\leftarrow$	root : system library files
*/tmp	$\leftarrow$	global: temporary, deleted on boot
*/dev/null	$\leftarrow$	root : trash bin (only write)
/dev/urandom	$\leftarrow$	root : entropy gen. (read only)
*/home/user	$\leftarrow$	user : User "home $\sim$" space
*/home(1-8)	$\leftarrow$	group : Extra mounts (personal workstations)
/home9/phyast	$\leftarrow$	root : General programs for Phys. and Astro.
/scratch	$\leftarrow$	group : Extra disk (personal workstation) non-backed-up

Shells

Bash or ksh vs Csh/Tcsh etc.

I use (and suggest) Bash

.profile ← Called on login shells

.*rc ← Interactive login

.logout ← Called on shell exit

Commands

basicCommands.sh, *AdvCommands.sh*, and cheat sheet.

Commands

basicCommands.sh, *AdvCommands.sh*, and cheat sheet.

man*	←	manual pages !!!!!
cd	←	toggle directory
pwd	←	current directory
ls	←	list
echo	←	repeat
export	←	save variables to environment
which*	←	locate program
time	←	runtime for programs
alias*	←	give alias to program (or instructions)
cat	←	read or concatenate files
cp	←	copy files
mv	←	move files
diff	←	difference between files
head/tail	←	read file forward/backwards
more	←	read file: page-like

Commands cont...

chmod* ← change modifier (permissions)
chown* ← change ownership (groups)
tar ← (un-)compress
find* ← find file/dir
sed ← file string search row (regex)
awk ← file string search col (regex)
grep* ← get string
git ← version control
ln ← link things
rsync* ← file transfer (w/ hashes)
scp ← Secure-shell copy
(s)ftp ← Secure-shell file transfer proticol
curl ← communicate via web
wget ← download from url
kill* ← terminate process
top* ← list process/hardware info

Commands cont...

ssh ← Secure-Shell protocol

↑ -XY destination *sends GUI

↑ good for quick login

vnc ← Remote Display

↑ good for extended login & to save work

I am “required” to tell you to contact casit@ou.edu anytime you want to install a program (even using pip)

I am “required” to tell you to contact casit@ou.edu anytime you want to install a program (even using pip)
That being said lets install without root access.

Accounts

1. Linux accounts are hosted on the phyast server.
2. Even though you may have a work station!

Accounts

1. Linux accounts are hosted on the phyast server.
2. Even though you may have a work station!
3. Standard space (8GB); with professor (16GB)

Accounts

1. Linux accounts are hosted on the phyast server.
2. Even though you may have a work station!
3. Standard space (8GB); with professor (16GB)
4. This doesn‘t include HDD/SSD space

Accounts

1. Linux accounts are hosted on the phyast server.
2. Even though you may have a work station!
3. Standard space (8GB); with professor (16GB)
4. This doesn‘t include HDD/SSD space
5. when coding there are 2 scenarios: account or computer environments
6. Guess which one you can change....

Environment

Dictates how things communicate and what process runs

printenv ← shows all environment variables

PATH ← Main variable that stores program location

LIBRARY_PATH ← Stores common libraries

- So if PATH stores program location, if we change it then voilà, we have new program
- We just need to put a program on the system...

Environment

Dictates how things communicate and what process runs

printenv ← shows all environment variables

PATH ← Main variable that stores program location

LIBRARY_PATH ← Stores common libraries

- So if PATH stores program location, if we change it then voilà, we have new program
- We just need to put a program on the system...
- Either get binaries (if you use mostly system programs)

or

install from source (I recommend this)

From Source

Extremely Easy... Follow *installingYourOwnProgram.sh*

From Source

Extremely Easy... Follow *installingYourOwnProgram.sh*

download ← of course download the program

configure ← "configure" it to your system

make ← compile program

PATH ← now set you path to point to it

From Source

Extremely Easy... Follow *installingYourOwnProgram.sh*

download ← of course download the program

configure ← "configure" it to your system

make ← compile program

PATH ← now set you path to point to it

DONE!

Automate This

(Home)Linuxbrew ← makers of Homebrew (Mac) bring you this package manager

"manual" ← Same process as before but use the *installprogram.sh* script provided

Linux in a nutshell Covered...

architecture ← The layout of the filesystem

commands ← Various common commands used

environment ← What is the environment and how to change it

Custom Programs ← Installing custom programs without root access

Questions?