BASIC LINUX CONCEPTS

UNIVERSITY OF THE
FREE STATE
UNIVERSITEIT VAN DIE
VRYSTAAT
YUNIVESITHI YA
FREISTATA

UFS
UV

GNU LINUX CONCEPTS

- GNU Linux and its kernel is Written in C
  - This means that C-concepts come heavily into play
  - One such concept is file and command case sensitivity:

    The following files can exist in the same directory and can all be different files:

    File1   file1   FILE1 filE1 ....

- Directories and files can't be named the same in the same directory level:

```
echo "New important file" > important_files
mkdir     important_files
mkdir: important_files: File exists
```

- Completion & Auto-Completion
  - Commands and paths can be autocompleted by tapping the TAB button
  - A very useful package to install

    bash-completion

GNU LINUX CONCEPTS

- GNU Linux uses POSIX file systems, which allows ACLs etc.
- Look at file permissions:

```
ls -l /etc/passwd
```

- Change a file's permission:

```
chmod  ug+x filename     #exec to user & group
```

- Change a file's owner (as root):

```
chown  user:group  file_name
```

- Special files exist, such as symbolic (aka soft) links:

```
ln -s /etc/passwd   password_file
```

- Although a file/directory name can contain a space....it is not recommended.

GNU LINUX FILE SYSTEM STRUCTURE

• /	#Root Directory ..... much like c:\
• /bin	#Most system wide executables like c:\windows
• /boot	#Grub boot partition with kernel images
• /etc	#System wide configuration files
• /home	#Holds users' home directories
• /lib(64)	#System libraries ..... much like dll files
• /mnt	#Can be used to mount external drives etc.
• /proc	#System devices (Hardware) and processes
• /root	#Root user's home directory
• /run	#Some temp files for services & processes
• /tmp	#Temporary files
• /usr	#Usually installation path of some apps/databases
• /var	#Files/databases/logfiles that change a lot

ENVIRONMENT VARIABLES

- System wide environment variables can be set in **/etc/profile.d/**
- User can use **~/.bashrc** to set profile after each login
- Important environment variables:
  - USER, HOME, PWD
  - PATH, LD_LIBRARY_PATH

- Can view variables using set, redirecting it to less:
  ```
  set | less
  ```
- Setting a variable:
  ```
  export PATH=$PATH:$HOME/bin
  ```
- Unsetting a variable:
  ```
  unset My_Variable
  ```
- Most HPC systems make use of environment modules to manage environment

GNU LINUX EXECUTABLES

- An application/script can be executed from anywhere if its path is in the PATH environment variable
- A file/script/application can be made executable using the **chmod** command
- You can see which executable is used (full path) using which:

```
which ls          #Note it may refer to an alias
```

- How to run a process in the background:

```
dd if=/dev/zero of=/dev/null count=6000000 &
```

- An application returns a return code after execution, and it is saved in the special environment variable **$?**
  - A value of 0 (zero) means it was successful
  - A non-zero value indicates that an error(s) occurred

```
cat /etc/passwd
echo $?
```

GNU LINUX EXECUTABLES

- Executing more than one command in a single line:

```
hostname; uptime; free -m; df -h /
```

- Execute a command only if the previous command was successful:

```
cd /var/log && tail dmesg
```

- Execute a command only if the previous command was successful, if it failed execute something else:

```
touch /etc/passwd && echo "done" || echo "failed"
```

REDIRECTING OUTPUT

- A useful feature is redirecting output to a specific device/file

```
ps -ef  > procs
```

- Another useful feature is redirecting output to a command
  - This is called piping, for you use the pipe character "|"

```
cat /proc/cpuinfo |  grep "flags" | sort -u
```

- One can also use the output within a command:

```
echo "Today's date is: $(date)"
```

- Special redirection of STDERR to a file:

```
strace ls   2> error
```

- Special redirection of STDOUT and STDERR to a file:

```
ls /* &> output
```

- Special redirection of STDERR to STDOUT:

```
ls /* 1>&2
```

UFS
UV

GNU LINUX BASIC TOOLS

- Information regarding a command (ls)

  ls --help; man ls; info ls; apropos ls
- Redirecting output….using pipe

  ls / | grep "m"
- Shell Scripting
  - Writing a shell script
- File/Directory permissions
  - Checking / changing file permissions
- Converting Windows files to Linux format

  dos2unix

  unix2dos
- Changing content in a file, using Regular expressions

  sed -i "s|search for this|replace with this|g" filename.txt

GNU LINUX BASIC TOOLS

- Connecting to other machines:
  - GNU Linux uses the SSH protocol

    ```
    ssh username@remote_host
    ```

  - If the username on the local machine is the same on the remote machine:

    ```
    ssh remote_host
    ```

  - If you just want to log in and execute a command on the remote host:

    ```
    ssh -n  username@remote_host "uptime"
    ```

TRANSFERRING FILES TO REMOTE HOSTS USING **SCP**

- Copy a file to a remote machine:

  scp this_file user_name@remote_host:

  – This will copy "this_file" to the home directory of the user
- Copy a file to a specific path:

  scp this_file user@remote_host:/tmp/

  – This will copy the file as /tmp/this_file on the remote host
- Copy a directory to a remote host:

  scp -r /etc/skel/ user@remote_host:/tmp

TRANSFERRING FILES FROM A REMOTE HOST

- Copy /etc/passwd from remote host to current directory:

```
scp remote_host:/etc/passwd .
```

- Copy a remote directory to a specific path on local:

```
scp -r remote_host:/etc/skel /tmp
```

GNU LINUX BASIC TOOLS

- Getting Software from the internet:
  - **curl, wget, dnf, git**
- Installing software
  - dnf install
  - rpm -ivh package.rpm
  - wget http://xxx; tar -zxf xxx;cd xxx; ./configure ; make; make install
- Searching for files
  - find / -name textfile.txt; locate
- Searching for content in files
  - grep -i “string inside a file” textfile.txt
  - Using Regular Expressions
    - grep “^LogFile=Log[0-9]*\.log$” textfile.txt

ACTIVITY

- Create a VM on a machine where you have access to
- Install GNU Rocky Linux on that VM
- Familiarize yourself with the bash environment
- Always log in with your own user, **don't become root unless you have to**
- Try installing and removing some packages using rpm and dnf

INSTALLING SOFTWARE

UNIVERSITY OF THE
FREE STATE
UNIVERSITEIT VAN DIE
VRYSTAAT
YUNIVESITHI YA
FREISTATA

UFS
UV

INSTALLATION OF SOFTWARE

- GNU Linux makes use of two types of software installations
  - The first is installing a binary – Precompiled Software
  - The second is installing from Source Code

- On a HPC system, we tend to install a precompiled package for most of the generic GNU Linux packages such as Apache, bind etc.
- When installing Scientific Applications, we try to install from source code, allowing us to compile software specifically for the hardware
- Another reason for installing from Source Code is to get the latest and greatest features from the packages

- Installing a binary is relatively simple
  - You can install using a package manager such as (dnf)
    - DNF searches for dependencies and (hopefully) takes care of all the dependencies
  - You can install using a low-level package installer (rpm)

INSTALLING SOFTWARE

- Prebuild Packages in RedHat systems follow the following convention:
  - <package-name>**-**<major ver>**.**<minor ver>**-**<release>**.**<RedHat Release>**.**<arch>
  - For instance:

    vim-enhanced-8.2.2637-20.el9_1.x86_64
- There are also development tools for some packages
  - These packages usually include files and source code that can be used to link against other applications:

    glibc-devel
- Finally, a lot of packages separate their libraries from the application. This allows other applications to make use of the libraries without the need to have the referring application installed:

  gettext-libs
- RedHat Linux also has Source RPMS that can be installed and used to compile your own RPM from what is called a SPEC file

UFS
UV

INSTALLING SOFTWARE FROM SOURCE

- The installation of software from source usually follows the following procedure:
  - Download the source code from a website
  - Extract the "tar ball" (tar -xvf package_name.tar.gz)
    - The source usually contains a README and INSTALL file which gives installation instructions
  - Perform the configuration (./configure --prefix=/usr/local)
  - Compile the source code (make -j 12)
  - Optionally, test the compiled code first (make test)
  - Install the software onto the system (make install)

LAB PRACTICAL

- Using dnf
  - Try to install "munge"
  - Install:
    - epel-release
    - munge
    - Midnight Commander
    - wget
    - A package that provides libpng16.so.16.37.0
    - Determine the dependencies for httpd
  - Try to remove but don't physically remove:
    - munge-libs

ADDITIONAL LAB PRACTICAL

- Execute:
  - export mirror_path=https://mirror.ufs.ac.za/rocky/9.2/
  - export app_path=$mirror_path/AppStream/x86_64/os/Packages
  - export base_path=$mirror_path/BaseOS/x86_64/os/Packages
  - wget $base_path/n/net-tools-2.0-0.62.20160912git.el9.x86_64.rpm
  - wget $app_path/h/httpd-2.4.53-11.el9_2.5.x86_64.rpm
  - wget $app_path/h/httpd-tools-2.4.53-11.el9_2.5.x86_64.rpm
  - wget $base_path/m/mailcap-2.1.49-5.el9.noarch.rpm

- Using the RPM command:
  - Install the downloaded net-tools package
  - Remove the munge-libs package without removing munge itself
  - Determine which package provides /etc/services
  - Determine which files the package nfs-utils installs
  - Determine if strace is installed
  - Install httpd
- Perform an ldd on the munge executable
- Remove munge using RPM