Introduction to Linux

Rakesh Raghavaraju & Sridhar Ravichandran
User Support Analyst
Centre for Advanced Computing
Queen's University
rs211@queensu.ca

October 18, 2024

Introduction to Linux Command Line

Rakesh Raghavaraju & Sridhar Ravichandran
User Support Analyst
Centre for Advanced Computing
Queen's University
rs211@queensu.ca

October 18, 2024

Centre for Advanced Computing

Compute Ontario

Queen's UNIVERSITY

About CAC

- Centre for Advanced Computing (CAC) at Queen's University supports the research community by providing access to digital research infrastructure (DRI)

- The CAC delivers high performance and highly secure computing, supporting hundreds of Canadian research groups, comprising thousands of researchers working in a variety of fields

https://cac.queensu.ca/

Digital Research Infrastructure(DRI)
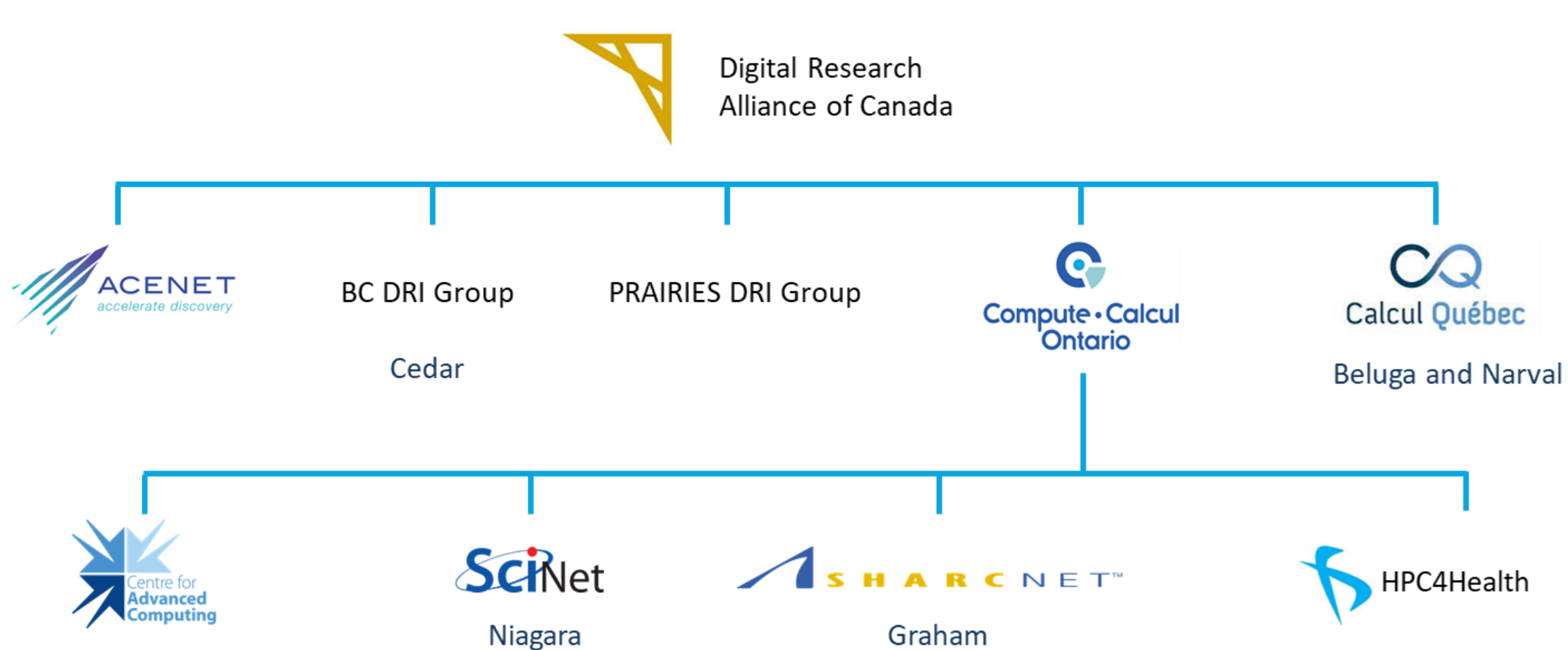
Overview of Linux Operating System
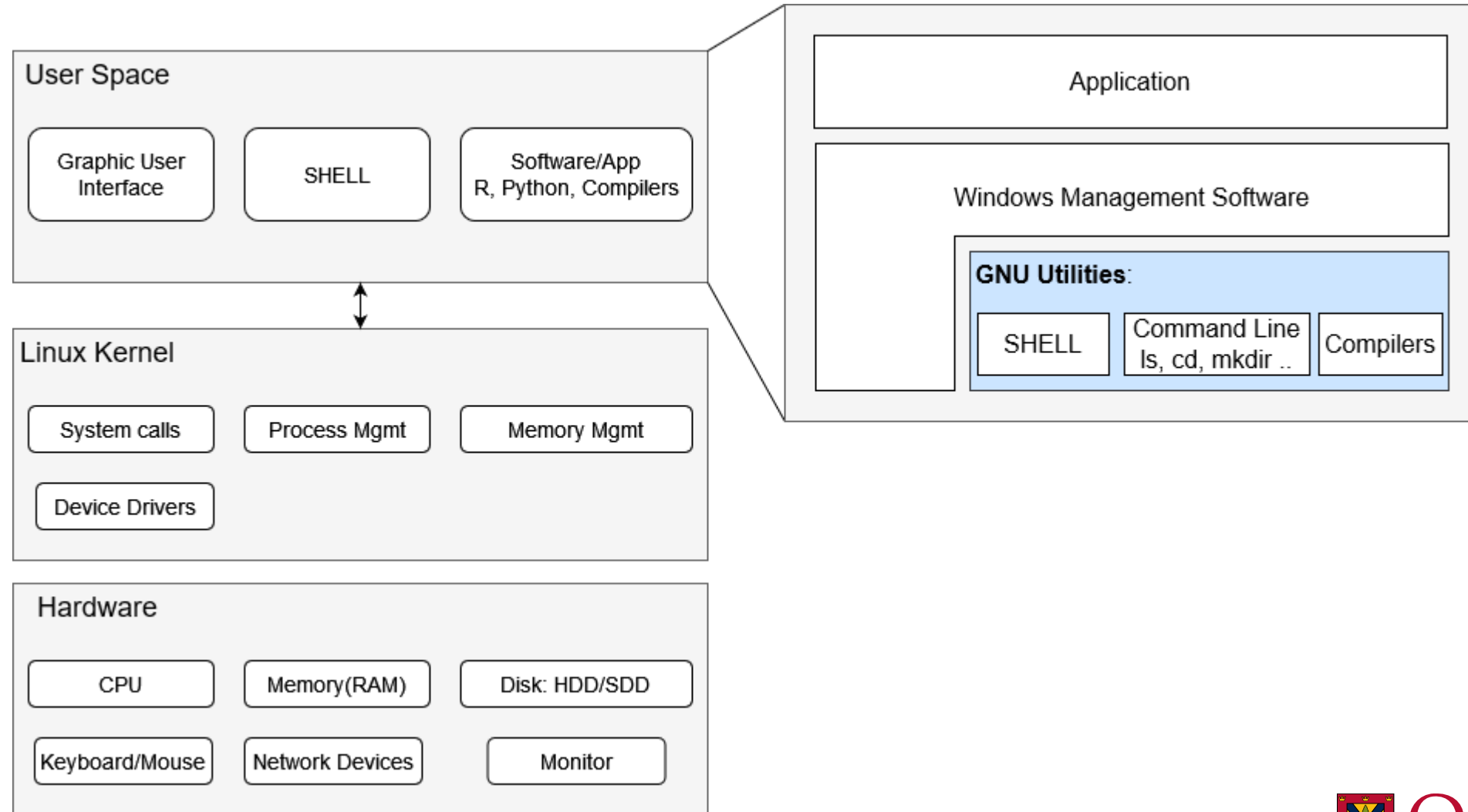
History

- 1970's : Unix OS was developed by *Denis Ritchie* and *Ken Thompson* at AT&T Bell Labs

- 1983 : **Richard Stallman** started the GNU Project to create a free and open-source version of UNIX

- 1987: **Andrew S. Tanenbaum** released MINIX, a UNIX-like system for academic use

- 1991 : **Linus Torvald** started working on project that later became the Linux Kernel using MINIX with GNU C compiler. Released in 1992 under GNU GP License

- 1993 on : Kernel was adapted to GNU environment and **several distros** were developed using **GNU/Linux**
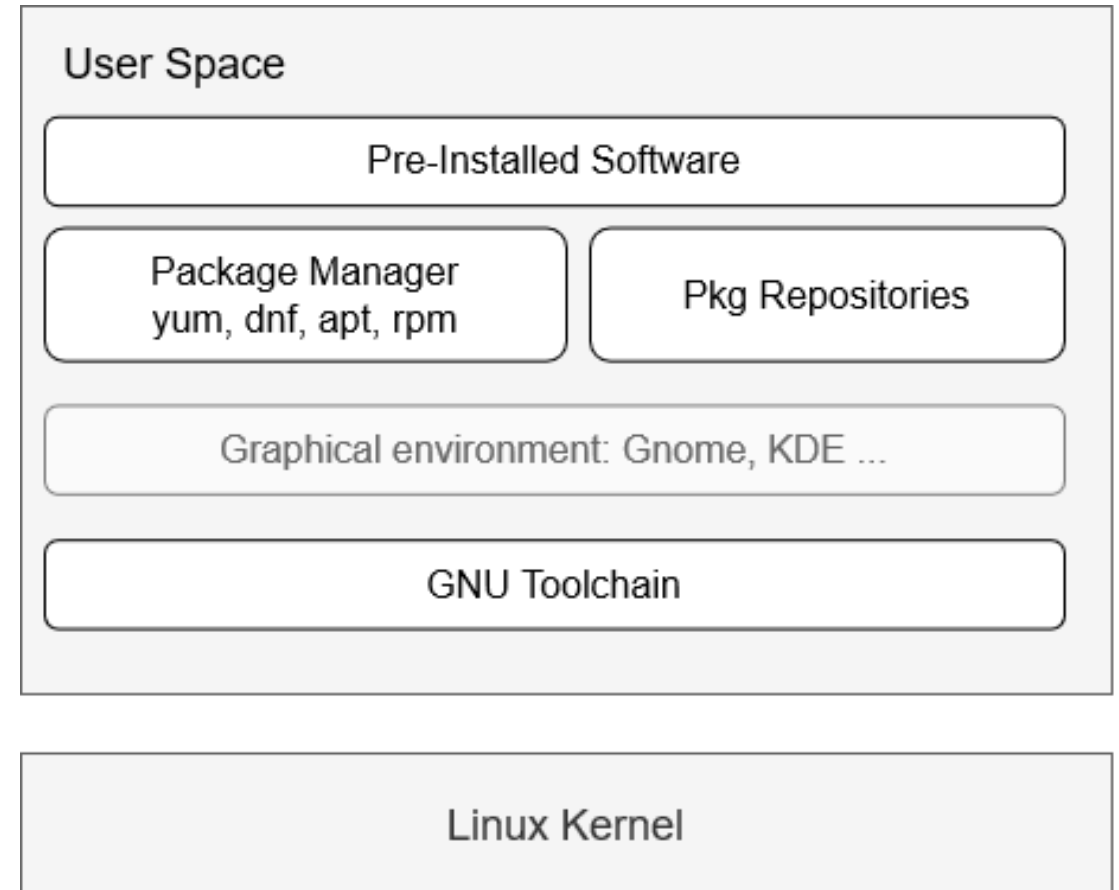

"I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things). "

Linux Distributions(distros)

A Linux OS that includes the Linux kernel, system libraries, utilities, a selection of pre-installed software, a package manager and in most case a graphic environment

- **Debian-based**: *Ubuntu*, Linux Mint, *Raspbian*

- **RHEL-based:** RedHat, CentOS, Rocky Linux, Fedora

- **SUSE:** OpenSUSE

- **Android:** based on the BusyBox (not GNU)

"Linux" distributions are distributions of GNU/Linux customized to a particular userbase

User Space
Pre-Installed Software
Package Manager yum, dnf, apt, rpm / Pkg Repositories
Graphical environment: Gnome, KDE ...
GNU Toolchain

Linux Kernel

GUI vs Command Line

Graphical User Interface

- Ease to get started, familiar interface

- Several pre-installed desktop application e.g., Libre Suite, Firefox

- Good at using existing functionality

- Flexibility: Less control and hard to do large and/or repetitive tasks

- Not ideal for accessing a remote server; rendering is quite slow with a high overhead; higher bandwidth usage

Command Line

- Hard to learn, you are given a blank canvas, and it can be overwhelming to navigate at first

- Commands that are efficient at doing that one thing

- Helpful in creating reproducible and repetitive workflows

- Commands can be saved(in a script) and can be used later

- Low overhead

- For the most part, only way to use HPC

Filesystem: What happened to my *C:* drive?
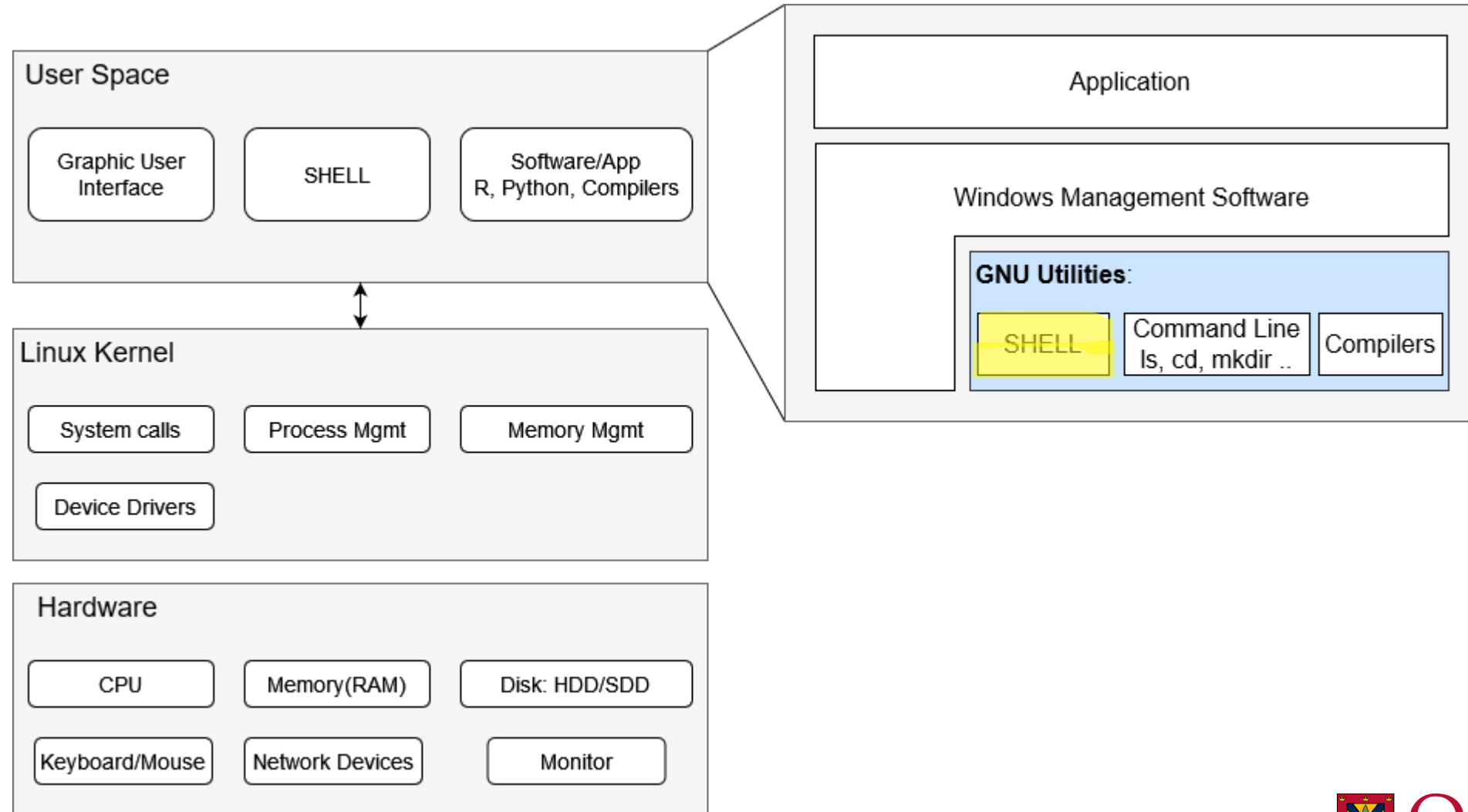
DEMO: Exploring Linux Filesystem with GUI

Filesystem
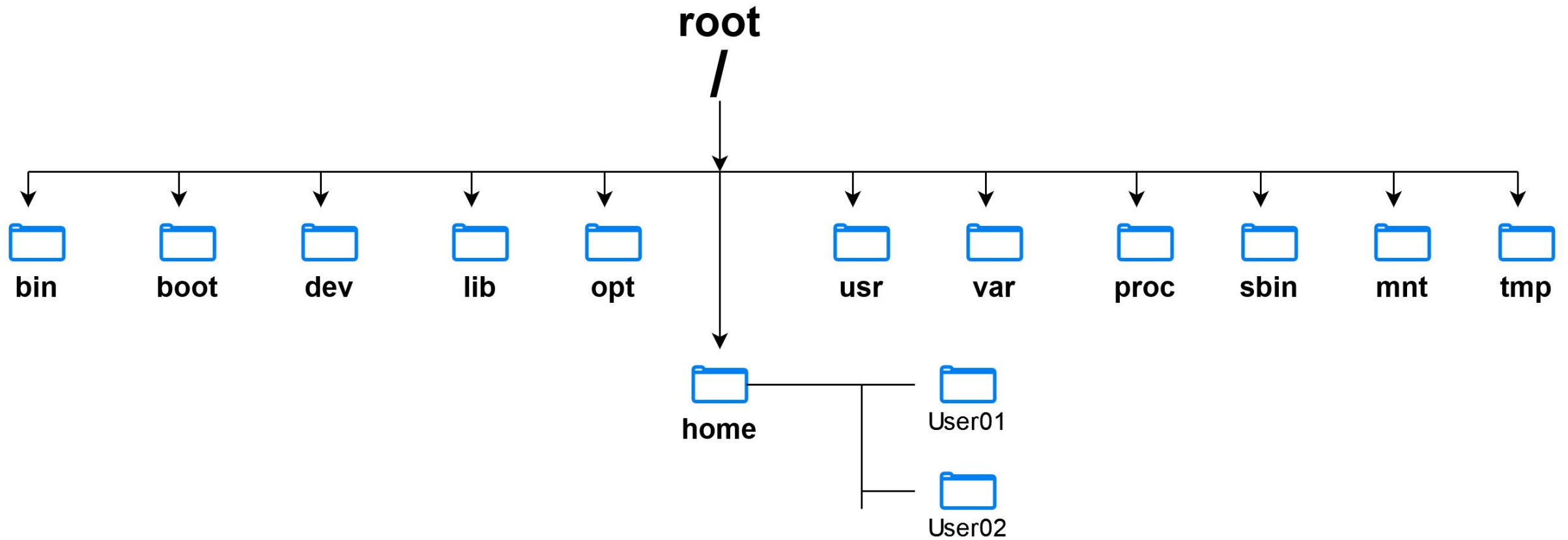
Filesystem

bin: Contains binaries(GNU Toolchain) which are program/app we can run

etc: System-wide configuration files

boot: Files required during the boot process. Should **NOT** be modified/touched

dev: Device files; enable interface between device and operating system

Sys: Virtual directory like /dev and contains info about connected devices

lib: Contains library files that user application and executables can use.

opt: For compile software and script that can be used system-wide

usr: Contains lib and bin directories and user applications are installed in this directory

var: Stands for 'variable' and contains files that vary with time, e.g., *log files, cache and backup*

proc: Contains system information such as info about CPU, Memory ...

sbin: Similar to /bin, but contains application that can only be used by 'root' (superuser) user

mnt: Mount storage devices

tmp: Temporary files

home: This is the directory for user's personal files; Typically, each users has a directory in this folder e.g., */home/rakeshr* or */home/sa105060*

Overview Linux Operating System

User Space

| Graphic User Interface | SHELL | Software/App R, Python, Compilers |

↕

Linux Kernel

| System calls | Process Mgmt | Memory Mgmt |

| Device Drivers |

Hardware

| CPU | Memory(RAM) | Disk: HDD/SDD |

| Keyboard/Mouse | Network Devices | Monitor |

Application

Windows Management Software

GNU Utilities:

| SHELL | Command Line ls, cd, mkdir .. | Compilers |

SHELL

- Shell is a super program

- It is an interface between the user and the operating system

- Almost all Operating Systems including Windows(Powershell) has a Shell program

- User can launch or run application using a Shell

- Terminal or Terminal Emulator(MobaXterm) to open a Shell program

- **Linux shell:** Bourne Shell (sh), Bourne Again Shell (**bash**), csh (C shell), ksh, ..

- **Bash** is most common shell, its default shell in most GNU based distros

Accessing Linux Shell

- **MobaXterm**

- **Terminal:** MacOS, Linux

- **Windows Subsystem Linux(WSL)**

- **Oracle VM VirtualBox**

- **Google Cloud Shell:** https://console.cloud.google.com/

- **HPC Cluster**: Frontenac

- **Locally installed Linux Distro**: Single board computer, Old laptop

Accessing the Frontenac cluster
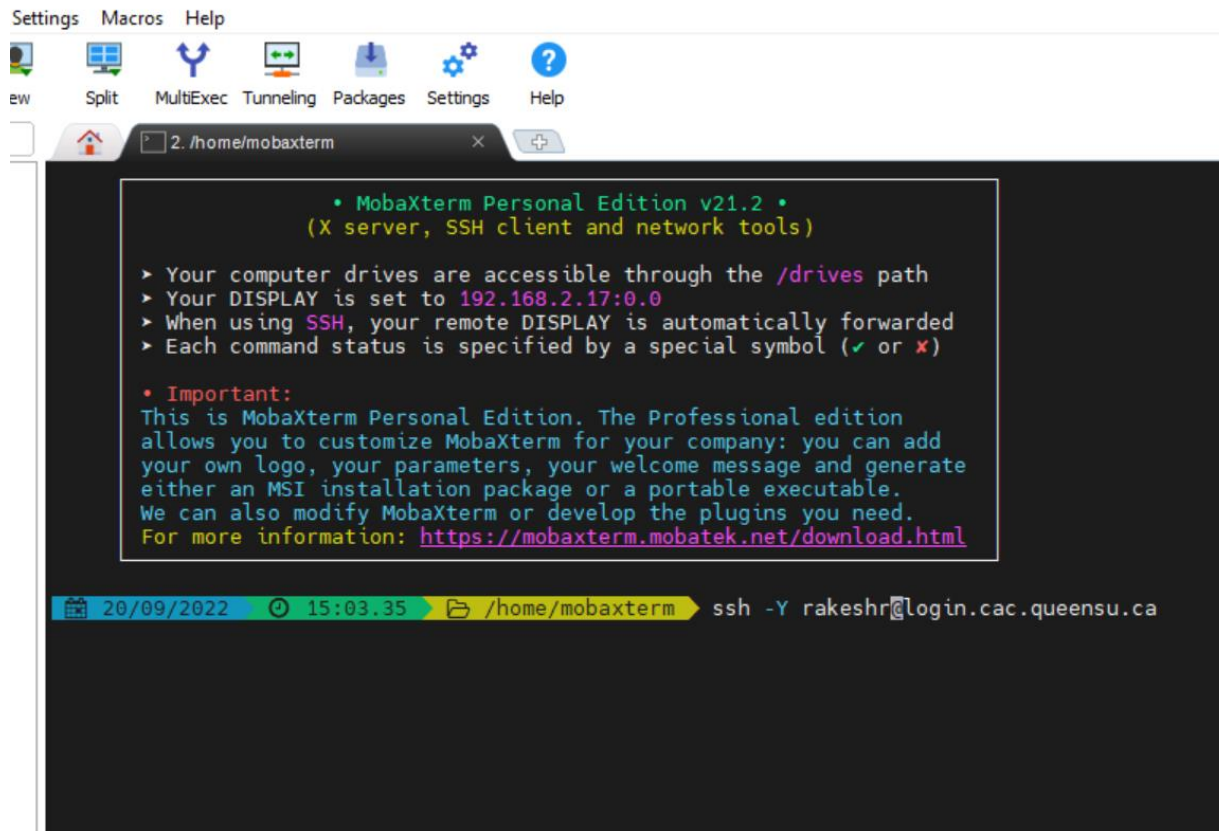
Requirements: SSH-Client to login

- MobaXterm, PuTTY for Windows

- Terminal on MacOS and Linux
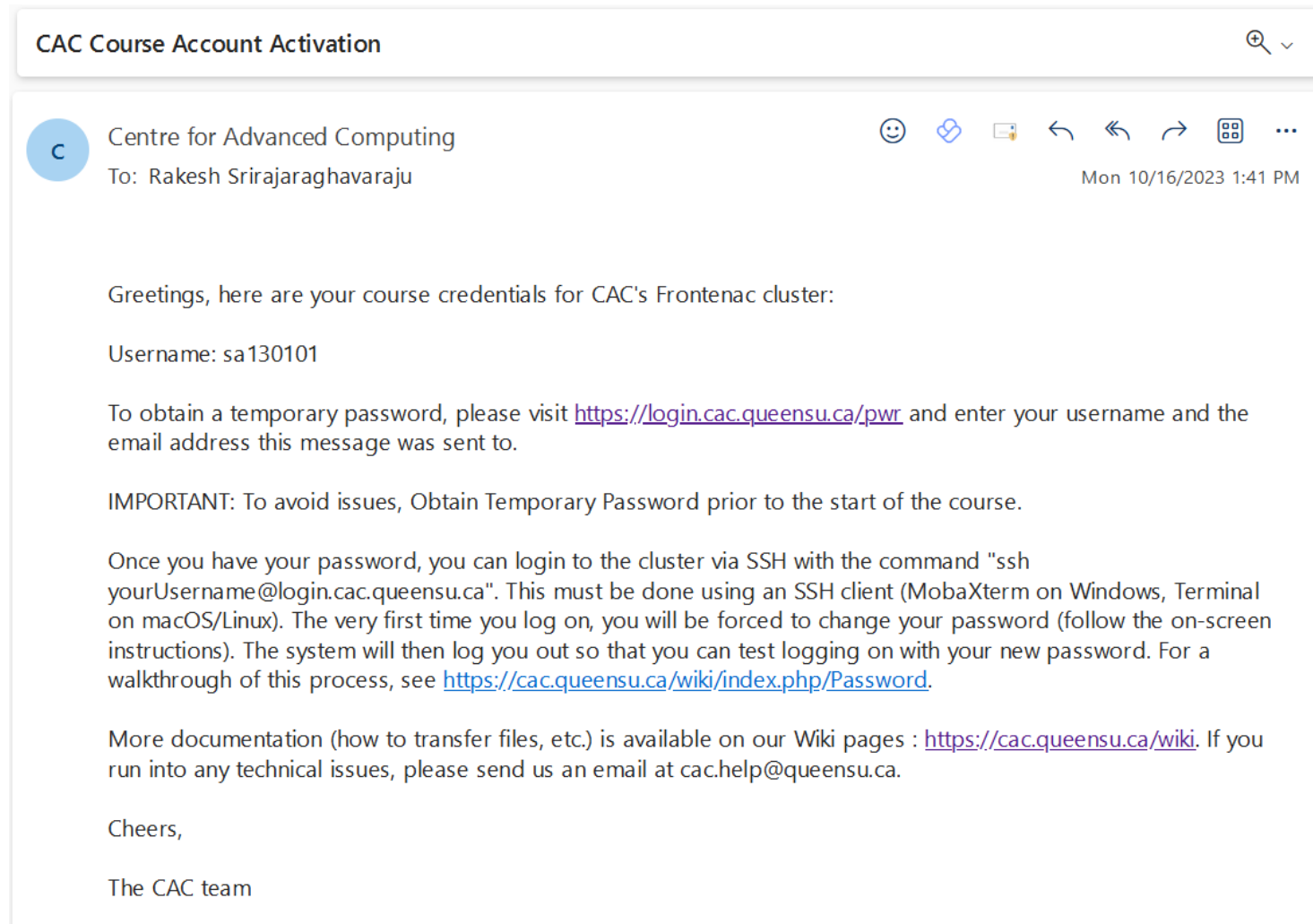
Username

ssh sa201xxx@login.cac.queensu.ca

Server Name

SSH stands for secure shell and enables a secure way to communicate between two hosts.

MobaXterm: https://mobaxterm.mobatek.net/download.html

PuTTY : https://www.putty.org/

Accessing the Frontenac cluster

CAC Course Account Activation

Centre for Advanced Computing
To: Rakesh Srirajaraghavaraju

Mon 10/16/2023 1:41 PM

Greetings, here are your course credentials for CAC's Frontenac cluster:

Username: sa130101

To obtain a temporary password, please visit https://login.cac.queensu.ca/pwr and enter your username and the email address this message was sent to.

IMPORTANT: To avoid issues, Obtain Temporary Password prior to the start of the course.

Once you have your password, you can login to the cluster via SSH with the command "ssh yourUsername@login.cac.queensu.ca". This must be done using an SSH client (MobaXterm on Windows, Terminal on macOS/Linux). The very first time you log on, you will be forced to change your password (follow the on-screen instructions). The system will then log you out so that you can test logging on with your new password. For a walkthrough of this process, see https://cac.queensu.ca/wiki/index.php/Password.

More documentation (how to transfer files, etc.) is available on our Wiki pages : https://cac.queensu.ca/wiki. If you run into any technical issues, please send us an email at cac.help@queensu.ca.

Cheers,

The CAC team

Exercise 0: Login to the cluster (10 mins)

1) Download or open a Terminal or Terminal emulator (MobaXterm)

2) Login to the Frontenac cluster

Navigating the filesystem

- Learn how to move around the filesystem through command line and explore the files

- Commands used

  - **pwd**   Present Working Directory

  - **ls**   list the files in the directory

  - **cd**   change directory

  - **man <command>**   Manual file for the command

```
[sa105060@caclogin03 ~]$ pwd
/global/home/sa105060

[sa105060@caclogin03 ~]$ ls
CAC_files.txt  jobname-5334773.out
jobname-%e.err       mpiinit-5345223.err  submitJob1.sh
large_submitJob.sh    mpiinit-5345223.out  submitJob.sh

[sa105060@caclogin03 ~]$ cd /
[sa105060@caclogin03 ~]$ ls
…
[sa105060@caclogin03 ~]$ cd ~
```

Navigating the filesystem

- cd command options

  '*cd – '* : Previous folder

  'cd ..' :   Change to parent directory

  'cd ../..' :   Two directories up to parent directory

  'cd ~': Change current working directory to 'home'

[GDB_Valgrind]$ ls -a
. .. 00_HelloWorld 01_readArg 02_Divisors 03_Factorial
04_CalcPi 05_InfiniteLoop 06_IllegalMemAccess
Debugging_PDO_March2023.pdf

Exercise 1: Exploring the filesystem on your local terminal

1) Open a terminal on your local computer(or on Frontenac) and change to $HOME directory and identify all the files in the folder

Exercise 1: Exploring the filesystem on your local terminal (10 mins)

1) Open a terminal on your local computer(or on Frontenac) and change to $HOME directory and

   identify all the files in the folder

   Note: Explore various flags for 'ls' command using 'man' or '- -help' and find the option to print all the files

   including hidden file(has a '.' prefix in the file name)

2) On Frontenac, change the directory to '/global' and list all the sub-folders

A few tips

- 'tab' will auto-complete the command

- Use ←,→ arrows to navigate within the command

- You can get previous commands using ↑ ↓ arrows

- 'history' to get a list of previous(~1000) commands used in the shell

- Run multiple commands in a single line

  - *command_1; command_2;*

  - *command_1 && command_2*
    # command_2 will only run after the command_1 is successful

Creating files

- Create files and learn how to read and write to the file/folder

- Commands that we will use

  - **mkdir**   Creates a new file in the

  - **touch**   Create an empty file

  - **echo**   Display text to screen/stdout

  - **nano**   Text editor

  - **cat**   Display the content of the file

```
[IntroLinux]$ mkdir Example01
[IntroLinux]$ cd Example01/

[Example01]$ touch file01.txt
[Example01]$ ls
file01.txt

[Example01]$ echo "First line of the file"
First line of the file
[Example01]$ echo "First line of the file" > file01.txt

[Example01]$ cat file01.txt
First line of the file

[Example01]$ nano file01.txt
```

Moving files

- Here, we learn on how move, copy, delete and create links to existing file

- Commands used

  - **mv**   move or rename a file

  - **cp**   copying files

  - **rm**   remove files

  - **ln**   create a link to a file

```
[Example01]$ ls
file01.txt
[Example01]$ cp file01.txt file01.bak.txt
[Example01]$ ls
file01.bak.txt  file01.txt
[Example01]$ rm file01.bak.txt
[Example01]$ mv file01.txt file01_renamed.txt
[Example01]$ mkdir backup_dir
[Example01]$ cp file01_renamed.txt file01.txt
[Example01]$ mv file01.txt ./backup_dir/
[Example01]$ ls
backup_dir  file01_renamed.txt
[Example01]$ ls backup_dir/
file01.txt
```

Exercise 2: Create files and folders (10-15 mins)

1) ~~Change directory to '/global/project/sg201000/IntroLinux' and create a folder with your username 'sa130xxx'~~

2) Create a file in this folder and enter your first line('This is the first line') into this file

3) Now copy the file to a new file

4) And add a second line to the folder 'This is a second line'

5) (Optional) Examine the difference between the first file and second file using 'diff'. Use 'man' command to find the syntax for diff command.

View file content

- Commands used

  - **more**   Creates a new file in the

  - **less**   Create an empty file

  - **head**   Display text to screen/stdout

  - **tail**   Text editor

```
[IntroLinux] $ cd grep_Tut
[grep_Tut] $  more NameList_200.txt
…
[grepTut]$ head -n 6 NameList_200.txt
FNAME, LNAME, City, Province
Arwen, Kirk, Ottawa, ON
Thor, Lannister, Kingston, ON
Trinity, Smith, Waterloo, ON
Athena, Gandalf, Winnepeg, MB
Juliet, Smith, Halifax, NS

[grepTut]$ tail -n 5 NameList_200.txt
Bilbo, Gandalf, Vancouver, BC
Trinity, Smith, Vancouver, BC
Shrek, Baratheon, Calgary, AB
Bruce, Loki, Halifax, NS
Athena, Bane, Toronto, ON
```

Examine file content

- Commands used

    - **grep**   search for a key word

    - **wc**   word count

- Wildcards

- Regular expressions (regex) special character

.	Match any single character
*	Any character must appear zero or more times
?	Preceding character must appear once
^	Pattern at the begin of the line
$	Pattern at the end of the line
\	Escape the special character
[]	Should contain any of the character
|	In pattern matching you can specify two or more patter

```
[IntroLinux] $ cd grep_Tut
[grep_Tut] $  grep 'Toronto'  NameList_200.txt

[grep_Tut] $ wc NameList_200.txt

[test0]$ wc -l example3.fastq
400 example3.fastq

[grep_Tut] $ ls ./Nek5000/bin/nekb*
nekb  nekbb  nekbmpi

[grep_Tut] $ ls ./Nek5000/bin/nek[1-9]*
nekb  nekbb  nekbmpi

[grep_Tut] $  grep '[tT]oronto'  NameList.txt

[grep_Tut] $  grep 'Toronto|Ottawa'  NameList.txt
```

Piping

- We can pipe(or re-direct) output from one command to another command using '|'

- You can find '|' just about the enter key on most keyboards

```
[IntroLinux] $ cd grep_Tut
[grep_Tut] $  cat NameList_200.txt |grep ON| wc –l
89
[grep_Tut] $ grep 'ON' NameList_200.txt | grep -v 'Toronto'  |wc –l
76
```

Exercise 3: Examine the content of the file (10 mins)

1) Copy '1_control_trnL_2019_minq7.fastq' from 'IntroLinux/CourseMaterial/grepTut' to your folder in 'IntroLinux /sausers/'

2) And count the number of lines in the file

3) (Optional) Find the number of lines with the sequence A,C,T,G

4) Download NEK5000 tar file from github and untar the file. You can find the commands in 'IntroLinux/CourseMaterial/grepTut DownloadCommands'

Searching for files

- Commands that we will learn

  - **find**   Find files and directories in a

    hierarchy tree

```
~ $ module load qt/5.9.6
~ $ cd EBROOTQT5
[5.9.6] $ find .  -type d -name ^Qt
...
[5.9.6] $ find .  -type d -name ^qt
```

Copying and Downloading files

- **wget** to download file from the weblink

- '**git clone**' to download file from github

- **tar** and **gunzip** to decompress the files

- **scp, sftp** commands to copy file to and from remote servers

```
[IntroLinux] $ cd grep_Tut
[grep_Tut] $ wget https://github.com/Nek5000/Nek5000/.../v19.0.tar.gz
...
[grepTut]$ ls -la v19.0.tar.gz
-rw-r----- 1 rakeshr wheel 4224008 Oct  9 22:48 v19.0.tar.gz

[grepTut]$ git clone https://github.com/Nek5000/Nek5000.git

[grepTut]$ ls -lad Nek*
drwxr-x--- 11 rakeshr wheel 4096 Oct  9 22:51 Nek5000
```

Copying files to the cluster

- Command line: rsync, scp

$ scp –R ./myfile.txt sa105060@login.cac.queensu.ca:/global/home/sa105060/Myfolder/

Recursive | Source path | Username:Server Addresss | Destination path

- Filezilla; SFTP session(MobaXterm): GUI interface

- Globus: For transferring large datasets

- References

https://cac.queensu.ca/wiki/index.php/UploadingFiles:Frontenac

https://docs.alliancecan.ca/wiki/Transferring_data

Globus: https://docs.alliancecan.ca/wiki/Globus
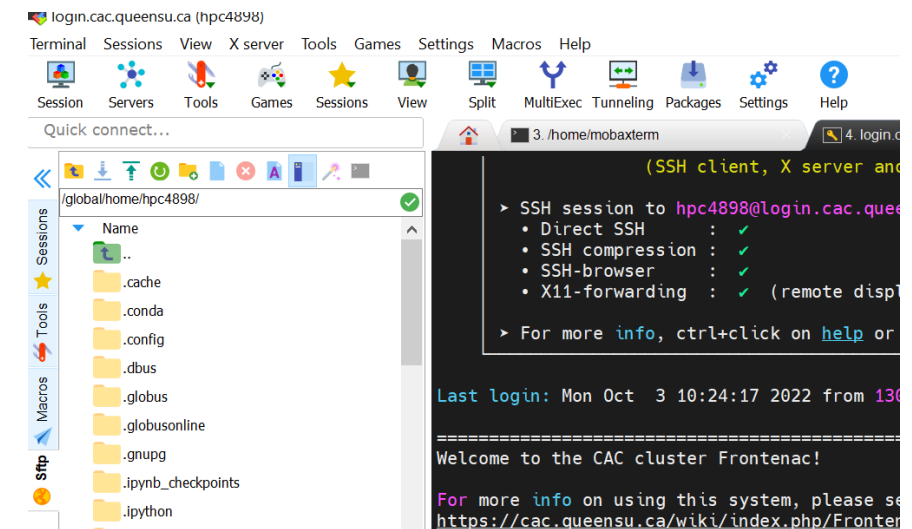
Exercise 4: Upload a file (10 mins)

1) Create a file called 'Local_file.txt'. Add a few lines to this file and upload it to Frontenac cluster

Redirecting input and output

- Output from the command is printed to screen; Input is streamed from the keyboard

- Redirecting stdin, stdout, stderr using '>' and '<'

>   to redirect the output to a file or device

<   to redirect input from a file or device to a command application

1>   stdout to a file or device

2>   stderr to a file or device

```
[04_CalcPi]$ ./calcPi
No arguments were entered defaulting to N=10000
The value of PI is: 3.1416

[04_CalcPi]$ ./calcPi > out.txt
[04_CalcPi]$ cat out.txt
The value of PI is: 3.1416

[04_CalcPi]$ ./calcPi 1>out.txt 2>/dev/null
[04_CalcPi]$ cat out.txt
The value of PI is: 3.1416

[04_CalcPi]$ ./calcPi 1>out.txt 2>&1
[04_CalcPi]$ cat out.txt
The value of PI is: 3.1416

[04_CalcPi]$ wc –l < calcPi.c
23
```

File permissions

d: directory
-: file
l: symbolic link

drwx rwx rwx

u:user **g**:group **o**:other

r: read
w: write
x: execute

Linux is a multi-user system and multiple users can use it, simultaneously. File permission are set to grant access to the user on the file system.

chmod: change file permission

$ chmod g+w Makefile

$ chmod u+x Makefile

$ chmod ugo+x Makefile

$ chmod 754 Makefile

4: read (r); 2: write (w); 1: execute (x)

chown: change ownership of the file

$ chown username:grpname filename

```
[simpleZeroCopy]$ pwd
/global/home/hpc4898/cuda-sample/0_Simple/simpleZeroCopy
[simpleZeroCopy]$ ls -la
total 515
drwxr-x---  3 hpc4898 hpcgtest 4096 Aug 31  2021 .
drwxr-x--- 41 hpc4898 hpcgtest 4096 Aug 31  2021 ..
drwxr-x---  2 hpc4898 hpcgtest 4096 Aug 31  2021 doc
-rw-r-----  1 hpc4898 hpcgtest 8236 Aug 31  2021 Makefile
-rw-r-----  1 hpc4898 hpcgtest 2111 Aug 31  2021 NsightEclipse.xml
-rw-r-----  1 hpc4898 hpcgtest  236 Aug 31  2021 readme.txt
-rw-r-----  1 hpc4898 hpcgtest 8731 Aug 31  2021 simpleZeroCopy.cu
```

Exercise 4: File permissions (10 mins)

1) Copy the folder from '/global/project/Workshop2023/IntroLinux/CourseMaterial/Example01' and troubleshoot if you face any issue

   Note: All *sa130XXX* accounts are part of *sg130000* group

Environment variables

- System variables:

  $PATH, **$LD_LIBRARY_PATH**, **$LIBRARY_PATH**

- User-defined variables

  export INSTALLPATH=~/bin

- Programs and Scripts can use **env** variable to obtain

  information about the system

```
[~]$ env
EBROOTR=/cvmfs/soft.computecanada.ca/easybuild/software/2020/avx2/Core/r/4.3.
1
HISTSIZE=1000
HOME=/global/home/rakeshr
HOSTNAME=caclogin03
LD_LIBRARY_PATH=/usr/lib64/nvidia:/global/home/rakeshr/.local/Software/v910/ru
ntime/glnxa64
LIBRARY_PATH=/cvmfs/.../R/lib:/cvmfs/.../r/4.3.1/lib:...:/cvmfs/.../lib/intel64
OLDPWD=/global/project/rakeshr/IntroLinux/RedirectingOutput/04_CalcPi

PATH=/cvmfs/soft.computecanada.ca/easybuild/software/2020/avx2/Core/r/4.3.1/bi
n::::/global/home/rakeshr/bin
PIP_CONFIG_FILE=/cvmfs/soft.computecanada.ca/config/python/pip-avx2-
gentoo.conf
PKG_CONFIG_PATH=/cvmfs/.../r/4.3.1/lib/pkgconfig:...:/usr/share/pkgconfig
PWD=/global/home/rakeshr
PYTHONPATH=/cvmfs/soft.computecanada.ca/custom/python/site-packages
SCRATCH=/scratch/rakeshr
SHELL=/bin/bash
USER=rakeshr
```

Exercise-5

- How does path command change with module load StdEnv/2016.4

- Get an interactive node using salloc and find all SLURM variables that start with 'SLURM'

Shell scripting

- Shell script is a set of commands stored in a file and can be re-used

- Command line is useful when you run 1 or 2 commands. But scripting become necessary for complex workflows

- Each shell script must start with '#!' line that specified the shell. Here we are using bash shell

  #!/bin/bash

- Commands in following lines are executed in sequence

- We can use conditional statement(if-else) and for loops in the script

Shell scripting

- Variable

- Passing arguments to the script

- Conditional statement

- For and while loop

Useful Linux commands

cd	Change directory
ls	List files in the directory
mkdir	Make directory
cat	Display the content
mv	Move/Rename a file
rm	Remove file/directory
history	List the past commands used
clear	Clear the terminal
cp	Copy files
echo	Print to screen
wc	Word count

file filename	Information about the file
which command	Location of the executable
diff file1 file2	Compare two files
wget	Download files from urls
sort file	Sort the lines
source file	Execute the command in the file
grep key file	Search for the 'key' in file
**cmd1	cmd2**	'|': Pipes the output one cmd1 as input for cmd2
tar	Untar/ tar files; similar to 'zip'
head –n 100 file	Prints first 100 lines
tail –n 100 file	Prints last 100 lines

Closing Exercise-6

- Copy this tar file to your local machine '/global/project/Workshop2023/IntroLinux/CourseMaterial.tar.gz'

- Unpack the files on your local machine.

Advanced commands

- SED

- AWK

References

1. Blum R and Breshnan C., 2015, Linux Command Line and Shell Scripting, 3rd Edition

2. Linux Foundation: https://training.linuxfoundation.org/training/introduction-to-linux/

        https://www.youtube.com/watch?v=sWbUDq4S6Y8&t=4552s

3. Brian Ward, How Linux Works: What every superuser should know

4. Wikipage: https://docs.alliancecan.ca/wiki/Linux_introduction

Thank you