TAMU users: If you’re outside campus, activate VPN by connect.tamu.edu
Computing Resources

The HPRC group currently administers four HPC clusters:

- ACES
- FASTER
- Grace
- Terra (retiring soon!)

You’ll need one of two options to use them:

<table>
<thead>
<tr>
<th>Credentials</th>
<th>Clusters</th>
<th>Who</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPRC Account</td>
<td>FASTER, Grace, Terra</td>
<td>Mostly Texas A&amp;M students/staff</td>
</tr>
<tr>
<td>ACCESS ID</td>
<td>FASTER and ACES</td>
<td>Researcher or educator at a U.S. academic, non-profit research, or educational institution</td>
</tr>
</tbody>
</table>

Link to our Knowledge Base: [https://hprc.tamu.edu/kb/](https://hprc.tamu.edu/kb/)
Your Login Password

• Do NOT share your password
• Do NOT share your account
• Texas law and TAMU regulations prohibit the sharing and/or illegal use of computer passwords and accounts
HPRC Portal

- HPRC webpage: hprc.tamu.edu
  - TAMU: portal-faster.hprc.tamu.edu
  - ACCESS: portal-faster-access.hprc.tamu.edu
HPRC Portal (ACCESS)

If you chose the second option on the previous slide, you’ll get the ACCESS CIlogon OpenID Connect page. Log-in using your ACCESS credentials. Create an account if you do not already have one.

Select the Identity Provider appropriate for your account.
Linux Using the Portal - Shell Access

Convenient shell access anywhere with a web browser

OnDemand provides an integrated, single access point for all of your HPC resources.

Message of the Day

IMPORTANT POLICY INFORMATION

• Unauthorized use of HPRC resources is prohibited and subject to criminal prosecution.
• Use of HPRC resources in violation of United States export control laws and regulations is prohibited. Current HPRC staff members are US citizens and legal residents.
• Sharing HPRC account and password information is in violation of State Law. Any shared accounts will be DISABLED.
• Authorized users must also adhere to ALL policies at: https://hprc.tamu.edu/policies

The terminal will ask you to log in again
Where Am I?

pwd command (print working directory)

Linux commands in green for you to type

```
pwd
```

command output in blue

```
/home/username
```

list contents of your working directory

```
ls
```
Navigating the Linux Directory Structure
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>mkdir my_dir</code></td>
<td><code>mkdir</code> to make a new directory</td>
</tr>
<tr>
<td><code>cd my_dir</code></td>
<td><code>cd</code> to change to another directory</td>
</tr>
<tr>
<td><code>cd ..</code></td>
<td><code>cd</code> back out of the current directory</td>
</tr>
<tr>
<td><code>rmdir my_dir</code></td>
<td><code>rmdir</code> to remove an empty directory</td>
</tr>
</tbody>
</table>
Directory Shortcuts

Linux has several special shortcuts to save you typing:

- = directory you’re in right now (the “current working directory”)
- = directory that contains the one you’re in now (“parent directory”)
~ = your home directory
- = the directory you were in before this one
Changing Directories: cd

Return to your home directory:

```
cd

cd ~

cd ~/

cd $HOME
```

Switch to the parent directory of the current directory:

```
cd ..
```

Return to previous directory:

```
cd -
```

```
cd $HOME
mkdir temp
mkdir temp/dir1
cd temp
pwd
cd dir1
pwd
cd ../..
pwd
cd ..../..
pwd
cd -
pwd
cd ..
pwd
cd ~
pwd
```
Absolute vs. Relative Path

If you are in the `project` directory

```bash
pwd
```
```
/home/chris/project
```

The relative path to the README file is `../docs/README`

```bash
ls ../docs/README
```

The absolute path to the README file `/home/chris/docs/README`

```bash
ls /home/chris/docs/README
```
Let's start working with content in our directories. Start with these basic commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>Writes file content on the standard output*</td>
</tr>
<tr>
<td>echo</td>
<td>Display a text string on the standard output</td>
</tr>
<tr>
<td>touch</td>
<td>Creates a new empty file</td>
</tr>
<tr>
<td>nano</td>
<td>Creates a new file or edit an existing file (text editor)</td>
</tr>
<tr>
<td>rm</td>
<td>Remove a file</td>
</tr>
</tbody>
</table>

Let's print some output and make a new file:

```
echo "Hello World"
touch new.txt
nano new.txt
cat new.txt
```

*Usually “standard output” just means your screen, but it can be moved
Using the Portal File Editor

In the “Files” tab in the portal

Create a new file and edit
History of Your Commands

Your commands are saved to a file in your home directory (.bash_history). You can use the up/down arrows to scroll through previous commands. Type `history` to see your previously entered commands.

```
history
history | tail
```

Search your command history using `|` and `grep`.

```
history | grep echo
```
Linux Commands Have Options

Leave a space between the command and the options

Spell out a full option with a double-dash:

```
ls --all
```

---all show all files, including hidden files which begin with ‘.’

Single dash lets you abbreviate:

```
ls -a -l
```

-a (shorter version of --all)
-l show file details

You can also combine (short) options behind one single dash:

```
ls -al
```

-a (same function as above)
-l (same function as above)

Remember directory shortcuts:

```
: current working directory
.. parent directory
```
Search for Linux Commands Options

Search the **manual** page for the Linux command  **ls**

```
man ls
```

- **f** move down (**forward**) one page
- **b** move up (**back**) one page

(Sometimes mouse scroll wheel and arrow keys work, too)

```
/all
```

- **n** search forward for **next** found match
- **N** search backwards next found match

```
/g
```

- **g** go to first line
- **G** go to last line

```
/q
```

quit
Linux Terminal Attributes

Depending on your terminal, you’ve probably been seeing different colors as you navigate.

File and directory names are colored based on their attributes such as permissions and extension (file type).

Note: These colors are not Linux-universal and can depend on the different terminal emulator or shell.
Changing Attributes: chmod

Set limits on who can modify files and directories with ‘chmod’

Follow the instructions at right to make some example files and check their details.

You should see a bunch of dashes and letters to the left. Those are the permissions.

1. To change the user’s permissions of file1.txt to read, write, execute:
   (will be -rwxrw-r--)
   
   chmod u+rwx file1.txt

2. To change the permissions of file2.txt to read and execute for all and write for the user:
   (will be -rwxr-xr-x)
   
   chmod 755 file2.txt
   (see next slide for what this number means)

3. To remove the execute permissions of file2.txt for all “other” users:
   (will be -rwxr-xr--)
   
   chmod o-x file2.txt
Changing Attributes: chmod

`chmod [options] [permission mode] [target_file]`

<table>
<thead>
<tr>
<th>Number</th>
<th>Permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No permission</td>
</tr>
<tr>
<td>1</td>
<td>Execute permission</td>
</tr>
<tr>
<td>2</td>
<td>Write permission</td>
</tr>
<tr>
<td>3</td>
<td>Write and execute permissions</td>
</tr>
<tr>
<td>4</td>
<td>Read permission</td>
</tr>
<tr>
<td>5</td>
<td>Read and execute permissions</td>
</tr>
<tr>
<td>6</td>
<td>Read and write permissions</td>
</tr>
<tr>
<td>7</td>
<td>Read, write, and execute permissions</td>
</tr>
</tbody>
</table>

- `u` = user
- `g` = group
- `o` = other
- `r` = read
- `w` = write
- `x` = execute
- `-x` = remove executable permissions
- `+x` = enable executable permissions

Note the permissions display format is `- uuugggooo`
Shell Script Exercise

A *script* will let you perform multiple commands at once.

We’ve created an example script, which you can copy and run yourself.

Navigate to your home directory

```
cd $HOME
```

Copy the script to your home directory

```
cp /scratch/training/spring_24_primers/my_script.sh .
```
Shell Script Exercise

View (or edit) the shell script

```
nano my_script.sh
```

make your shell script executable

```
chmod 755 my_script.sh
```

run your shell script

```
./my_script.sh
```

```bash
#!/bin/bash

# HPRC shell script exercise

my_var="People"

echo "Howdy $my_var" > output.txt

mkdir script_output

mv output.txt script_output

cd script_output

cat output.txt
```
Shell Script Explanation

The “shebang”; all bash scripts must have this at the very top so the computer knows how to run it.

Shebang:

```
#!/bin/bash
```

Shebang (continued):

```
# HPRC shell script exercise
```

```
my_var="People"
```

```
echo "Howdy $my_var" > output.txt
```

```
mkdir script_output
```

```
mv output.txt script_output
```

```
cd script_output
```

```
cat output.txt
```

Pound signs start comments. They’re for you to leave notes; the computer doesn’t do anything with them. (The shebang is the exception!)

```
echo "Howdy $my_var" > output.txt
```

```
mkdir script_output
```

```
mv output.txt script_output
```

```
cd script_output
```

```
cat output.txt
```
Shell Script Explanation

```
#!/bin/bash
# HPRC shell script exercise
my_var="People"

echo "Howdy $my_var" > output.txt
mkdir script_output
mv output.txt script_output
cd script_output
cat output.txt
```

A “variable.” Call later with ‘$’ to reuse stored data.
The ‘>’ redirects the output to the filename you provide.
(Commands we’ve seen previously)
Shell Script Exercise

View (or edit) the shell script

```
nano my_script.sh
```

make your shell script executable

```
chmod 755 my_script.sh
```

run your shell script

```
./my_script.sh
```

```
#!/bin/bash
# HPRC shell script exercise
my_var="People"
echo "Howdy $my_var" > output.txt
mkdir script_output
mv output.txt script_output
cd script_output
cat output.txt
```
Exit your terminal

exit

exit the terminal session

To fully logout of the FASTER portal, you need to exit the browser.
Thank you

Any questions?
Need Help?

First check the FAQ [https://hprc.tamu.edu/kb/FAQ/Accounts/](https://hprc.tamu.edu/kb/FAQ/Accounts/)
- Email your questions to help@hprc.tamu.edu

Help us help you -- provide the following info:
- Which cluster you’re using
- Your username
- Job id(s) if any
- Location of your jobfile, input/output files
- Application used, if any
- Module(s) loaded, if any
- Error messages
- Steps you have taken, so we can reproduce the problem
Continued Learning

Intro to HPRC Video Tutorial Series

HPRC’s Knowledge Base