The Hellgate Cluster is a Command Line Interface (CLI) Linux Operating System, this tutorial will introduce some basic useful Linux commands to help you navigate and create files.

**Connecting to Hellgate**

Hellgate is a headless server meaning it does not have a Graphical User Interface (GUI). Therefore, the recommended way of connecting to Hellgate is done through SSH (Secure Socket Shell), allowing for remote login and command-line execution. Here are multiple ways of creating SSH sessions, here are a few applications offered by Operating Systems.

**Windows**

**PowerShell**

A cross-platform task automation solution that includes a command-line, scripting language and a configuration management framework.

This comes pre-installed on a majority of Microsoft Window devices.

**Windows Terminal**

A modern, powerful and productive terminal application that integrates Command Line, PowerShell and Azure Cloud Shell. Some benefit of Terminal is its ability to open multiple tabs and customizable features.

This can be installed through the Microsoft Store.

**MacOS / UNIX / Linux**

**Terminal**

A powerful command-line interface that allows users to interact directly with the operating system’s Unix-based core.

This comes pre-installed on a majority of Apple, UNIX and Linux OS.

**Signing-in to Hellgate**

To sign-in to the Hellgate cluster, users will need to start a SSH session to the login node using the user’s preferred terminal:

ssh <NetID>@login.rci.umt.edu

This will prompt the user for a password. (NOTE: This will not show characters being typed)

By default, the User will be put in their home directory: /mnt/beegfs/hellgate/home/<NetID>

**tmux**

tmux is a powerful terminal multiplexer that’s widely used on High-Performance Computing (HPC) systems. It allows you to create and manage multiple terminal sessions from a single screen. This can be incredibly useful when working on remote systems like HPC clusters because it provides the ability to:

Some of tmux's features:

* Detach and reattach to terminal sessions, allowing users to continue running jobs even after disconnecting.
* Multiple window management within a single session.
* Allowing persistent sessions between devices.

**Command Format**

Interacting with the Hellgate cluster is done through commands, here is a breakdown of a typical command.

**Breakdown**

When on a UNIX Command Line Interface, you will be presented with a prompt such as:

[<NETID>@login ~]$ ls

This may seem like a lot of information but is simple to disassemble. Let’s start by splitting the prompt into two sections, a device portion and a command portion.

[<USER>@<HOSTNAME> <LOCATION>]$ <COMMAND> -<OPTION> <ARGUMENT>

**Device Portion**

This part of the prompt states information related to the connection.

[<USER>@<HOSTNAME> <LOCATION>]

* <USER> This represents the username of the person logged into the system. It’s the account under which the current session is running. On Hellgate, this will be your NetID.
* @ This separates the username from the hostname.
* <HOSTNAME> This is the name of the computer or server which the user is currently logged in to. On the Hellgate cluster, this will often be ‘login’.
* <LOCATION> This indicates the current working directory, where in the file system the user is located. On the Hellgate cluster, it will often only show the directory the user is located in.

**Command Portion**

$ <COMMAND> -<OPTION> <ARGUMENT>

* $ The prompt character, this indicates the shell is ready to take commands. ($ signifies a standard user, while # signifies a root user)
* <COMMAND> The command the user would like to execute.
* -<OPTION> The hyphen precedes an option or flag that modifies the behavior of the command, this is optional but can be very useful. Each command has their own options which can be found by consulting the manual, also known as [man pages](https://www.notion.so/HPC-New-User-Orientation-0fae82ea9b304c1da4054a78b3e01707?pvs=21). Often options are chained together.
* <ARGUMENT> Arguments are additional pieces of information or data the command operates on, such as file names, variables or inputs.

**Example**

A breakdown of a command.

[ab123456@login ~]$ tail -n 20 example.txt

**Device Portion**

* Username: ab123456 (NetID)
* Hostname: @login
* Location: ~ (/mnt/beegfs/hellgate/home/ab123456)

This shows user ab123456 is signed into Hellgate’s Login Node and is at location ~ (Their home directory).

**Command Portion**

* Command: tail
  + tail is used to display the last 10 lines of a file.
* Option: -n 20
  + -n lets tail know to display a specified number of lines, in this case, it is 20 lines of the file, instead of the default 10.
* Argument: example.txt
  + ab123456 is specifying to use tail on example.txt.

**Man Pages / —Help**

Short for manual pages, they provide extremely useful information on how to use a command and available options for the command.

man <command>

Often a combination of options (signified with a - or —) can help provide more or less information.

If a man page does not exist for the command, try using —help.

<command> --help

**Navigating Hellgate**

Learning to navigate without a Graphical User Interface (GUI) can be frustrating, here is a layout of how Hellgate’s directories are configured and some commands to navigate around.

**Hellgate Cluster Directory Layout**

Hellgate’s layout is not too complicated:

## USER DIRECTORY LAYOUT - HELLGATE CLUSTER ##

/mnt/beegfs/

├── projects/ ## Projects directory recommend for storing data

│ └── <NetID>/ ## 10TB of storage

│ └── file1 ## Absolute path: /mnt/beegfs/projects/<NetID>/file1

├── scratch/ ## Scratch directory is faster for processing data

│ └── <NetID>/ ## 5TB of storage

│ └── file2 ## Absolute path: /mnt/beegfs/scratch/<NetID>/file2

└── hellgate/

└── home/

└── <NetID>/ ## 500GB of storage - Login directory

Signing-in will place the user in their home directory: /mnt/beegfs/hellgate/home/<NetID>

**Command Line Interface Navigation**

**pwd**

To familiarize yourself with where you are, the pwd command will print your working directory, listing the absolute path of your location in the Hellgate cluster. An absolute path is a detailed path referenced from root, allowing commands to know exactly where to look for something.

pwd

This would output something like /hellgate/home/<NetID>

**ls**

To list content in the current directory, use ls.

ls

There are many options ls takes, many of them are extremely useful for finding information.

Listing content in Long format listing (-l) and human-readable (-h), providing more information about content. This is a very common command to use, therefore we have set an alias of ll to be short for ls -lh.

ls -lh

On Unix systems, hidden files will start with a period. These are usually hidden as most users will not interact with these files. The -a flag will display these files.

ls -a

**cd**

To change directories, use the cd command. TAB can be extremely useful in auto-completing long names.

cd example\_directory

You can also specify absolute paths, which must start from root (/).

cd /mnt/beegfs/projects/$USER

To move outside of a directory (up one level)

cd ..

To move to the previous directory you were just at. (Useful if using absolute paths)

cd -

To move to your home directory (Default is set to /hellgate/home/<NetID>)

cd ~

**Directories on Hellgate**

Directories (also known as folders) are used to organize files on the cluster, they can contain files and other directories.

**Creating Directories**

mkdir, short for make directory, will create a new directory in your current location.

mkdir example\_directory

You can specify absolute paths to create directories elsewhere.

**Removing Directories**

rmdir, short for remove directory, will delete an existing empty directory in your current location.

rmdir example\_directory

If the directory has files within it, you can use rm -r to recursively remove all content within the directory and then remove the directory.

rm -r example\_directory

**Moving Directories**

Sometimes directories are created in the wrong spot or you would like to reorganize, directories can be moved using mv. Often it is easiest to specify an absolute path to move the directory to.

mv example\_directory /mnt/beegfs/projects

**Copying Directories**

Directories can also be copied with cp -r to recursively copy all content in the directory.

cp -r example\_directory another\_directory

You can also specify an absolute path to copy the directory to.

cp -r example\_directory /mnt/beegfs/projects

**Files on Hellgate**

Often it is easier to troubleshoot, read, modify and alter files on the Hellgate cluster, rather than downloading the files back onto a user’s computer then re-uploading it.

**Reading Files**

Reading files on the Hellgate cluster can be done a few ways.

**cat**

Cat, short for concatenate, will write all the content of the file onto the terminal screen. Be cautious with large files.

cat example\_file

**less**

Less creates an interactive session, displaying only a screen’s worth of the file’s content from the beginning of the file, this is extremely useful for large files.

Users can scroll using the up and down arrows or using page-up and page-down.

less example\_file

To exit less, press q.

**tail**

The tail command is used to display the last lines of a file. By default it prints the last 10 lines of the specified file to the terminal.

tail example\_file

Tail can be used with option -n to specify the number lines to display.

tail -n 5 example\_file

**Writing Files**

Editing files without a Graphical User Interface can be extremely frustrating due to using hot-keys and key combinations instead of clicking buttons.

**touch**

Touch will create an empty file or update the access and modification timestamps of an existing file.

touch example\_file

Touch is optional, as nano, vi and vim will create a file if it does not exist.

**nano**

We recommend using nano, as it is the most user-friendly text editor.

nano example\_file

This will throw you into an interactive session with some characters at the bottom of the terminal, the ^ represents CTRL.

* To save the changes made to the file, use CTRL+w
* To exit nano, use CTRL+x. If you have not saved changes, it will prompt your to either press Y to save, or N to not save.
* If you get stuck in an unknown option, try using CTRL+c to cancel the option.

**Removing Files**

Removing files can be done with rm.

rm example\_file

**Moving Files**

Files can be moved with mv. Often it is easiest to specify an absolute path to move the file to.

mv example\_file /mnt/beegfs/projects/$USER

mv can also be used to rename files.

mv example\_file renamed\_file

**Copying Files**

Much like moving files, copying can also be achieved on the cluster.

cp example\_file example\_copy

You can also specify absolute paths to copy files elsewhere.

cp example\_file /mnt/beegfs/projects

**Moving Files to/from Hellgate (SFTP)**

We recommend using SFTP to upload (put) and download (get) data onto and from the cluster. SFTP allows is a Secure File Transfer Protocol which allows for remote file sharing. We recommend starting SFTP from the researcher’s device, as it is complicated to find the address of the researcher’s device, but easy to find the Hellgate Cluster’s address.

**SFTP Commands**

**Establishing A Connection:**

Much like SSH, you must login to the cluster in order to establish a connection through a terminal such as PowerShell:

sftp <NetID>@login.rci.umt.edu

After successfully log-in, your terminal prompt should appear like:

sftp>

To exit SFTP:

sftp> exit

**Navigation:**

Navigating the Hellgate cluster is very similar to ssh:

List files on the remote device:

sftp> ls

Moving around on the remote device:

sftp> cd remote\_directory

You can also list files on the local device:

sftp> lls

And move directories on the local device:

sftp> lcd local\_directory

**Upload / Download**

SFTP is picky and requires either absolute paths or to be inside the directory containing the files.

To upload files onto the Hellgate Cluster: (You can also upload directories)

sftp> put local\_file

To download files from the Hellgate Cluster: (You can also download directories)

sftp> get remote\_file

**Scripts and Sbatch**

Scripting is a programming technique which enables the automation of tasks that would otherwise be performed manually. A major benefit of scripting is the ability to save commands into a file, allowing users to easily re-run multiple commands using a single command.

**Creating and Using Scripts**

**Creating a Script:**

Scripts on the cluster will usually be in the form of a shell script (.sh), we can easily make a script using nano.

nano example\_script.sh

In the script, we’ll put commands which will run when we call the script.

#!/bin/sh

VAR\_EXAMPLE="example"

echo "This is an $VAR\_EXAMPLE"

Let’s break down this script:

* #!/bin/sh: A shebang, this tells the system which interpreter to use on the script.
* VAR\_EXAMPLE: A variable, this can change be used to reference values using $ in front of the name.
* echo: A command, the script will run this command when called

# is used to comment on the script, the interpreter will not read this. This can help provide information about what a script does or to disable a command from running.

**Using a Script:**

sh example\_script.sh

**Helpful Tips**

Some helpful tips with using the Command Line Interface.

**History**

Terminals keep track of used command and these commands can be recalled using the up and down arrows, saving you from re-typing the command. You can also view a list of previous command using history, to limit the output, try piping it to tail. ( the | character is typically above the enter bar )

history | tail

**Tab**

Tab will attempt to auto-complete arguments for you, saving a lot of time and preventing typos. If there are similar matches, it will auto-complete until the arguments differ. Pressing tab twice will provide a list of possible arguments.

**Cancelling Commands**

Often you will find yourself stuck in a command with no way to exit, CTRL+c will send a cancel signal to the program and return you to the command prompt.

**Alias**

Aliases in Bash are shortcuts for commands or sequences of commands. They allow you to create custom, easy-to-remember names for longer or more complex commands, making your command-line work more efficient.

**Creating an Alias**

To create an alias, use the following syntax:

alias example\_alias=’echo “this is an alias”’

This will create an alias that lasts for the current session and can be used if you type example\_alias into the command line.

**Making Aliases persist**

To make aliases permanent, add them to your ~/.bashrc file. The .bashrc file is a script that Bash loads whenever you start a new shell session.

nano ~/.bashrc

After adding the alias, you will need to exit the shell or reload ~/.bashrc

source ~/.bashrc

**Wildcard**

Wildcards represent any character, meaning they can be used to specify multiple files. In UNIX systems, they are represented with an asterisk (\*).

Say I would like to list all content in a directory, but only text files.

ls \*.txt

This specifies to use ls on all files as long as it ends with .txt.

An applicable example would be if we have a directory full of files with a consistent naming scheme, such as data11-2024.csv, data12-2024.csv, data01-2025.csv, data-02-2025.csv

If we wanted to move only data from 2024 to a new directory, we could use a wildcard to help use quickly automate this.

mv \*2024.csv ./2024-data

This would move anything ending with 2024.csv (data11-2024.csv and data12-2024.csv) to directory /2024-data and leave all other files (data01-2025.csv and data02-2025.csv).

**Putting It All Together**

Here is the typical usage on Hellgate for users.

Log-in to the Hellgate Cluster:

ssh <NetID>@login.rci.umt.edu

Find the script:

ls # look at files in current directory

cd # move directory

Modify a script:

nano sbatch.sh

Run the research program (Typically a sbatch script):

sbatch sbatch.sh

Verify job is running by looking at your current jobs:

squeue —me

If there is an issue with the Sbatch script, it will typically fail within a minute.