Linux

An Introduction

Mahmoud Parvizi, Ph.D.
Research Consultant, Institute for Cyber-Enabled Research
Research Associate, Department of Physics and Astronomy
parvizim@msu.edu

Linux

An Introduction

The term **Linux** is generally shorthand for the GNU/Linux Operating System

- Open-source, general-purpose operating system (OS)
- Developed in the 1990's to rival a costly OS like Unix or MS-DOS
- Quickly adopted by NASA and public universities
- Now used in mobile devices, personal and supercomputers

Linux

An Introduction

A **Linux distribution (distro)** is OS software components bundled together in accordance with the needs of the user

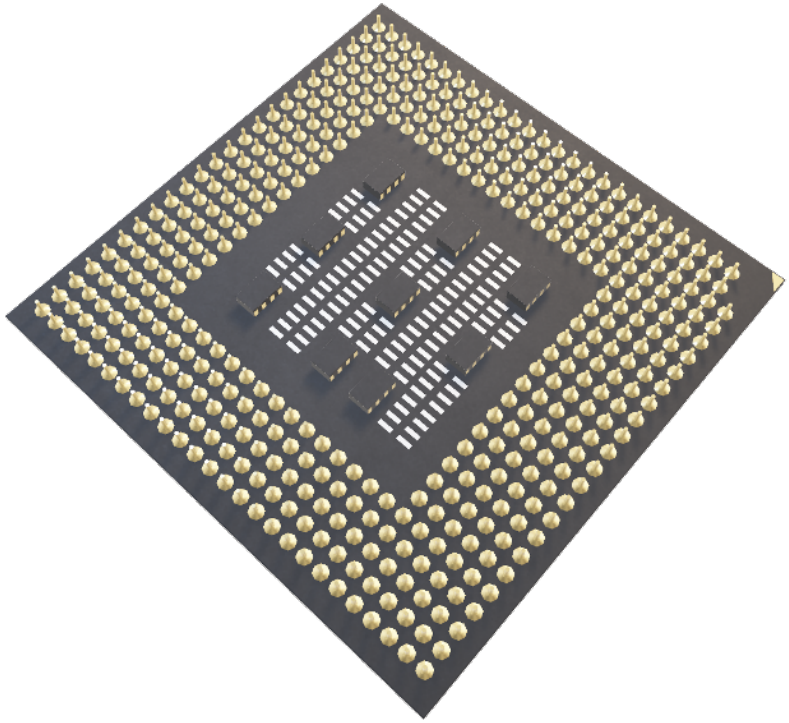
- Typically Includes:
  - Linux core
  - GNU tools and libraries
  - X window system
- Over 1,000 distros available
- Common free and open-source distros include Ubuntu and CentOS

Linux

An Introduction

The **Linux Core** (AKA ‘kernel’) is the OS software in control of the computer hardware

- 1991, created as free replacement for proprietary Unix OS; kernel.org



- Allows for optimized:
  - parallel computing
  - memory management
  - local/remote filesystems

Linux

An Introduction

GNU Tools and Libraries is a standard collection of OS programs used by the user installed programs

- 'GNU' is a recursive acronym;
  "GNU is Not Unix"
- 1983, GNU Project developed as free and open-source Unix compatible software
- Example: GNU C Compilers (GCC) for translating C, C++ code

Linux

An Introduction

The **X Window System** is software that manages the display

- Framework for drawing and moving graphics on a remote display
- ‘X’ specifically designed for network connections
- X.org Foundation leads free and open-source ‘X Project’

Linux

An Introduction

Three components for interaction between user and hardware

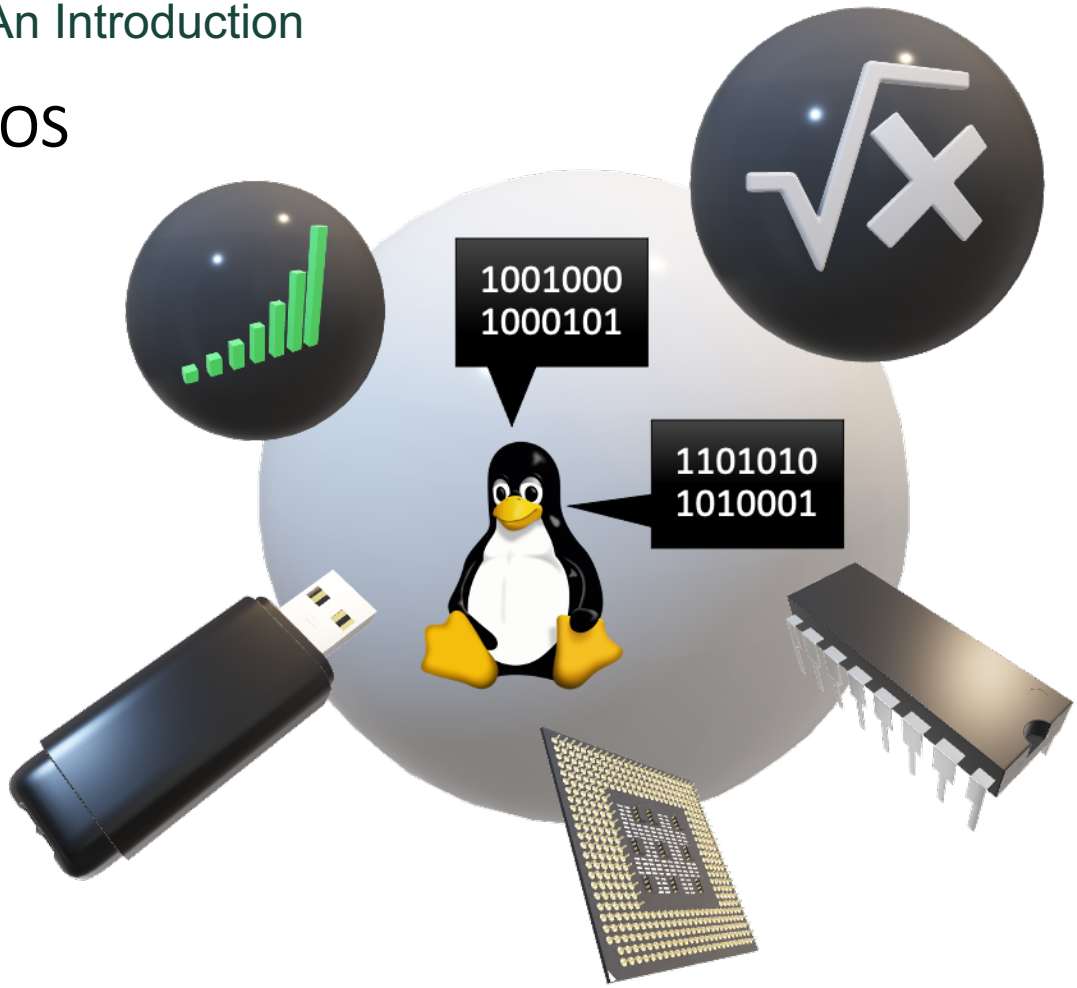
1. **Kernel**
2. **Shell**
3. **Terminal Emulator**

Linux

An Introduction

The **kernel** is the core of the OS

- Runs as machine code or ‘binary language’ layer
- Links instructions from software to computer hardware
- Manages the processors, memory and peripheral devices

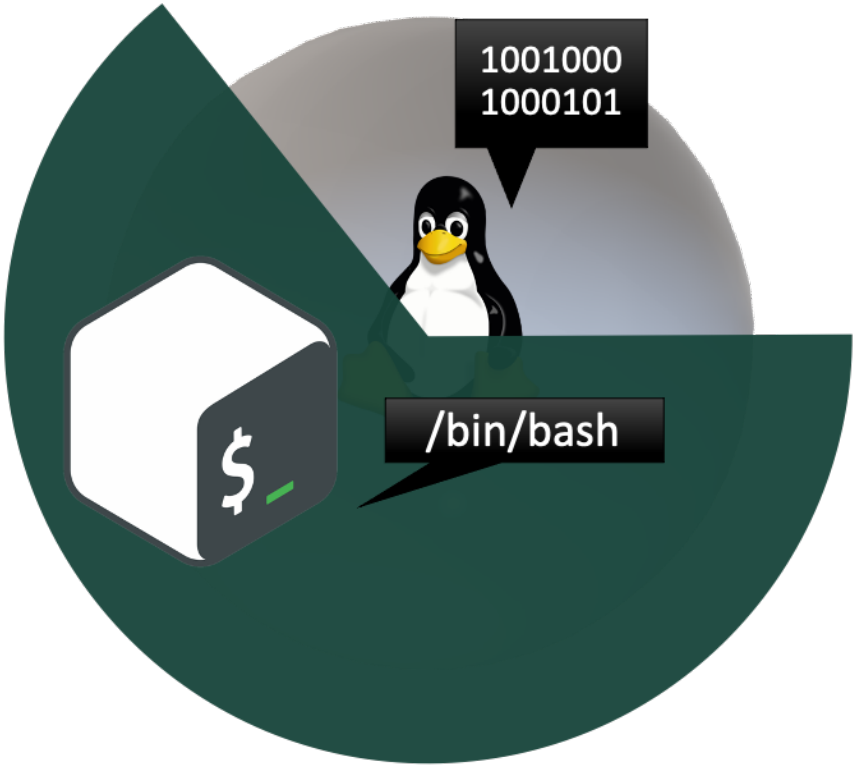


Linux

An Introduction

The **shell** is the outer layer of the OS

- Manages instructions from user to kernel
- Both an interactive and scripting language
- 1979, Bourne Shell (sh) developed for easy scripting
- 1989, Bourne-Again Shell (bash) added easy interactive use for GNU OS



Linux

An Introduction

The **terminal emulator** application gives the user access to the shell

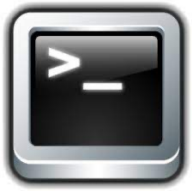
- Software in combination with keyboard, mouse and display
- Shell access may be local or remote *e.g.*, MSU HPCC
- Emulator may be ‘text terminal’ or graphical user interface (GUI)
- We will emulate a text terminal and use the **command line interface** (CLI)

Linux

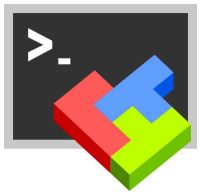
An Introduction

ICER Recommends these terminal emulators w/ X Windows

- **macOS:**
  - **Terminal.app** (included w/ macOS)
  - **Xquartz** (installation required)



- **MS Windows:**
  - **MobaXterm** (installation required)
  - **X Server** (included w/ MobaXterm)



Linux

An Introduction

macOS: Launch the emulated text terminal (AKA '**Terminal**')

- Use the **Finder** app

- Open the **Applications** folder

- Open the **Utilities** folder

- Double-click **Terminal** app

Linux

An Introduction

macOS: Launch the emulated text terminal (AKA '**Terminal**')

- You will see the CLI as a window, with prompt:
  `user@computer ~ $`
- Enter 'commands' in CLI to access the shell

Linux

An Introduction

macOS: Check for the **X Window System**

- Enter **xeyes** at the command line *i.e.,*

```
user@computer ~ $ xeyes
```

- Check for the ‘eyes’ window and the ‘X’ icon in menu bar

Linux

An Introduction

macOS: Check for the **X Window System**

- **If** you get the error message…

Error: Can’t Open Display

- Download and install ‘Xquartz.app’ from Xquartz.org

Linux

An Introduction

macOS: Verify the **Xquartz** installation

- Reenter **xeyes** at the command line *i.e.,*

```
user@computer ~ $ xeyes
```

- Check for the ‘eyes’ window and the ‘X’ icon in menu bar

Linux

An Introduction

Windows: Download and Install [MobaXterm Home Edition (HE)]

- Download **MobaXterm HE - 'Installer Edition'** from: mobaxterm.mobatek.net
- Right-Click to 'Extract All' from **MobaXterm_Installer.zip** folder
- Open **MobaXterm_Installer**
- Double-click **MobaXterm_Setup**

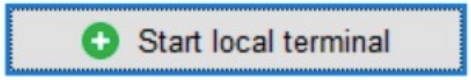
Linux

An Introduction

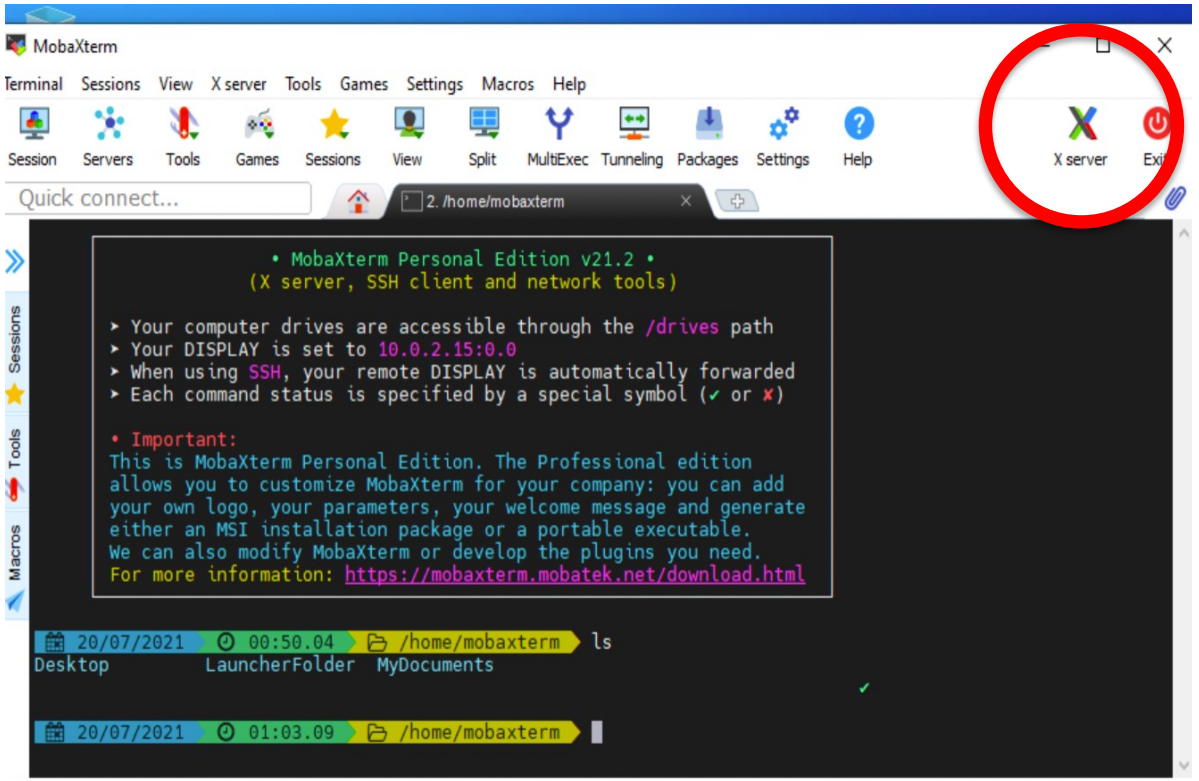
Windows: Launch [MobaXterm Home Edition (HE)]

- Double-click **MobaXterm** desktop icon

- Click on button



- Comes with **X Server** installed



Contact ICER

https://icer.msu.edu/contact

ICER
INSTITUTE FOR CYBER-ENABLED RESEARCH

Linux

The Command Language

Mahmoud Parvizi, Ph.D.
Research Consultant, Institute for Cyber-Enabled Research
Research Associate, Department of Physics and Astronomy
parvizim@msu.edu

Linux

The Command Language

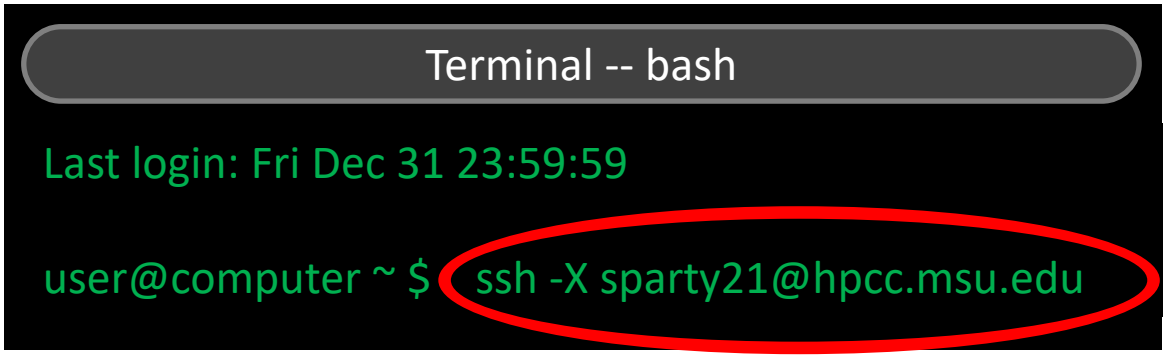
Linux is the operating system (OS) for the supercomputer at MSU's high-performance computing center (HPCC)

- User inputs commands through **terminal**; a remote command line interface (CLI) to OS **shell**
- Shell passes these instructions to OS **kernel** that then tasks computer's hardware

Linux

The Command Language

The **Command Language** is the syntax employed by user to administer tasks to be passed from shell to kernel



- Shell is command language interpreter; bash default on HPCC
- Programming language used interactively and in scripts

Linux

The Command Language

The bash interpreter reads the characters input in the terminal's CLI and groups them into **tokens**

- Sequences of characters interpreted as a unit; separated by a **blank**, *i.e.* 'space' or 'tab'
- May be a single or multiple character sequence

Linux

The Command Language

Tokens are then categorized as either a **word** or an **operator** depending on the constituent characters

Word: sequence of ordinary characters *e.g.,* **a** *,* **echo** or **my_file.txt**

Operator: sequence of special characters with a specific purpose *e.g.*, **&&** and **||** as logical *and* and *or*

Linux

The Command Language

Words DO NOT contain **metacharacters** *

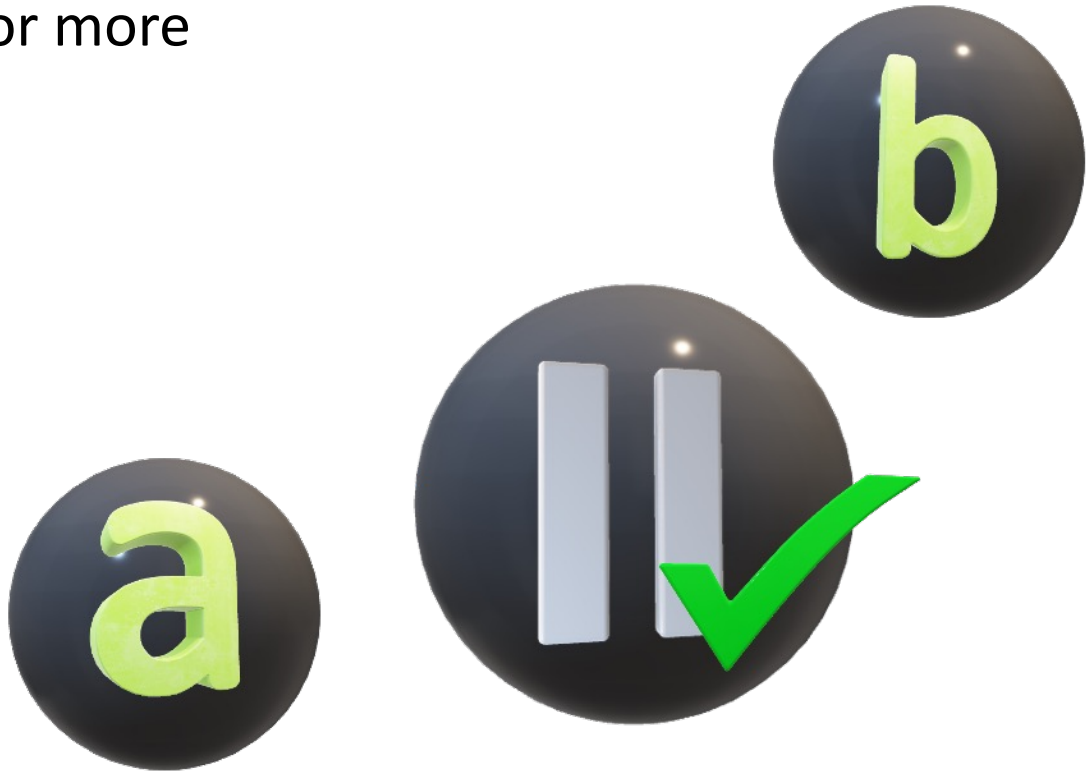
- Blanks *i.e.,*
  'space' *and* 'tab'
- 'newline'
- **&** , **|** , **;** , **(** , **)** ,
  < , and **>**

Linux

The Command Language

Operators contain at least one **metacharacter** *

- Sequence of one or more metacharacters
- Examples:
  - **;**
  - **| |**
  - **>&**
  - ‘newline’



* Excludes blanks

Linux

The Command Language

A sequence of words forms a **simple command**

Terminal -- bash

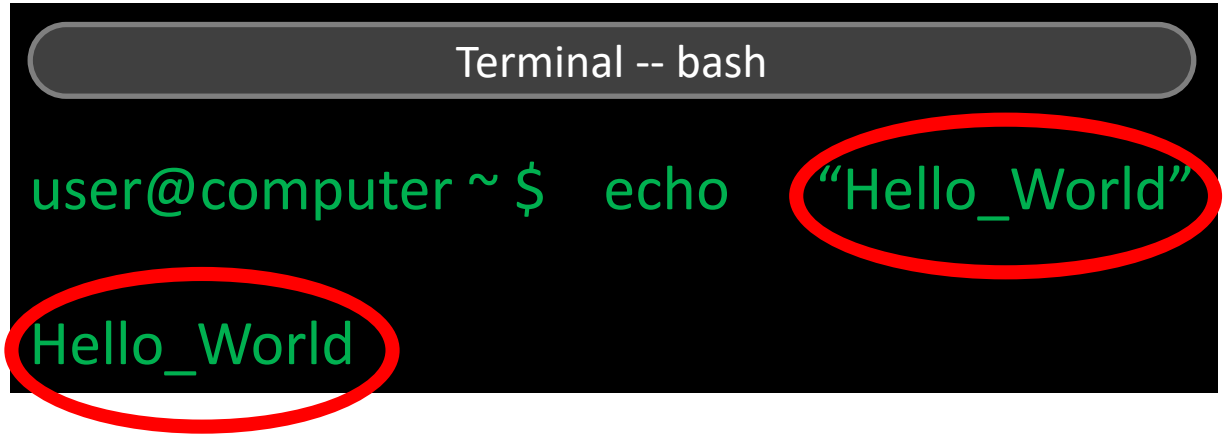
```
user@computer ~ $ ssh -X sparty21@hpcc.msu.edu
```

- Command to be executed → `ssh`
- Command Option(s) → `-X`
- Command's Arguments → `sparty21@hpcc.msu.edu`

Linux

The Command Language

A **command** uses ‘standard streams’ for input and output of data



- **stdin:** default is argument given at CLI

“Hello_World”

- **stdout** default is write to the display

Hello_World

Linux

The Command Language

The **command** to be executed may either be **builtin** or **external**

- **Builtin:** command executed internally by the shell *e.g.*, **echo**

```
user@computer $ echo “Hello_World”
```

- **External:** An executable program located elsewhere in the system *e.g.*, **python**

```
user@computer $ python -c “print(‘Hello_World’)”
```

Linux

The Command Language

A command **option** (AKA flag or switch) modifies the operation of the command

```
user@computer $ ssh -X sparty21@hpcc.msu.edu
```

- Identified by **-** or **--** then a letter or keyword
  e.g., **- h** or **-- help**
  depending on command

- Letters may be combined for brevity *e.g.*,
  - a - b ⟷ - ab ⟷ - ba

Linux

The Command Language

Command **arguments** are the constructs used as command input

```
user@computer $ ssh  -X sparty21@hpcc.msu.edu
```

- May be no arguments or multiple arguments *e.g.*,

```
sparty21@hpcc $ logout
```

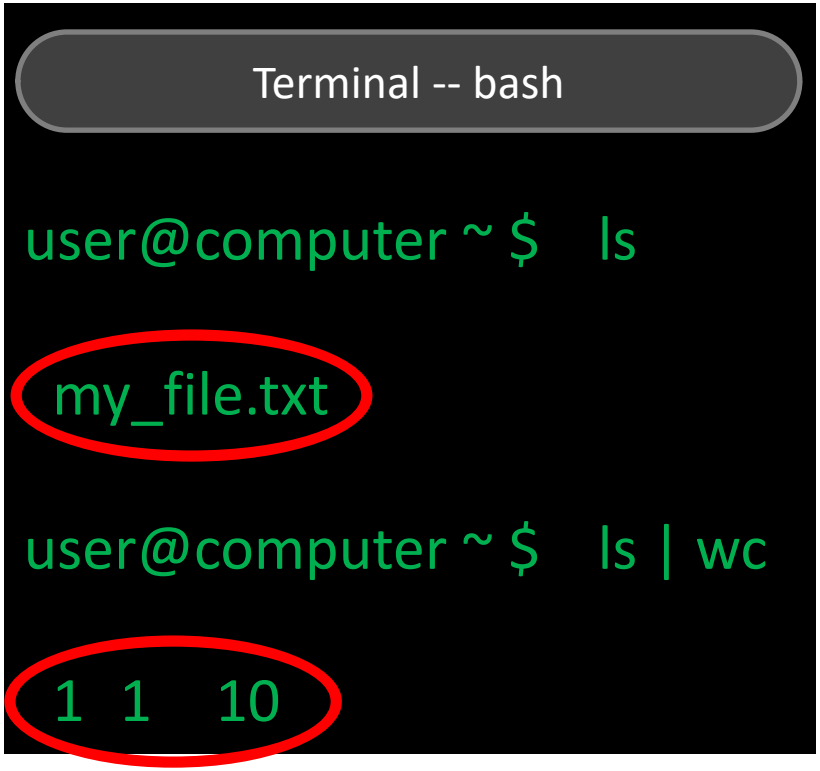
```
user@computer $ echo “Hello” “World”
Hello World
```

Linux

The Command Language

A **pipeline** is sequence of simple commands separated by **|** operator; output from former is input to latter

- Piped output from **ls** (list) command: '**my_file.txt**' not displayed
- Input to **wc** (word count) command
- Output from **wc** displayed: **1 newline, 1 word, 10 bytes**

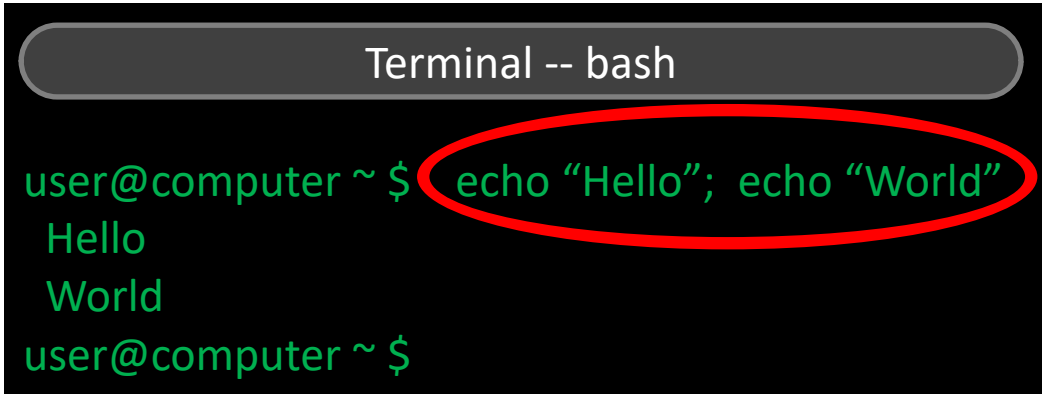


Linux

The Command Language

A **list** of commands is a sequence of simple commands or pipelines separated by the **;** , **&&** , or **| |** operator

- **;** executes commands sequentially
- **&&** conditional on successful execution of previous
- **| |** conditional on unsuccessful execution of previous



Linux

The Command Language

Redirections, or changing command stdin/stdout from default, carried out by operators **<** and **>** respectively

```
user@computer $ echo “Hello_World” > my_file.txt
```

- **>** creates a file named **my_file.txt** containing output **Hello_World**

```
user@computer $ echo “I_Am_Here” >> my_file.txt
```

- **> >** appends the output **“I_Am_Here”** to the file **my_file.txt**

Linux

The Command Language

An **error** occurs when a command fails to execute successfully; streams message to standard error (**stderr**)

- **stderr:** Error message stream writes to display by default

```
user@computer $ ssh “Hello World”
ssh: Could not resolve hostname hello world: Name or service not known
user@computer $
```

- May be redirected using **>&** to **my_error.txt**

```
user@computer $ ssh “Hello World” >& my_error.txt
user@computer $
```

Linux

The Command Language

To **group** a list of commands, use the **(** and **)** operators

```
user@computer $ ( echo “Hello”; echo “World” ) > my_file.txt ;
echo “I_Am_Here” >> my_file.txt
```

- First creates a file named **my_file.txt** containing output **“Hello World”**
- Then appends the output **“I_Am_Here”** to the file **my_file.txt**

Linux

The Command Language

Learning the bash commands

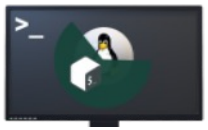
- Remember from training

- Ask your lab mate

- Ask the web

- Consult docs.icer.msu.edu

- Use the OS **manual pager**

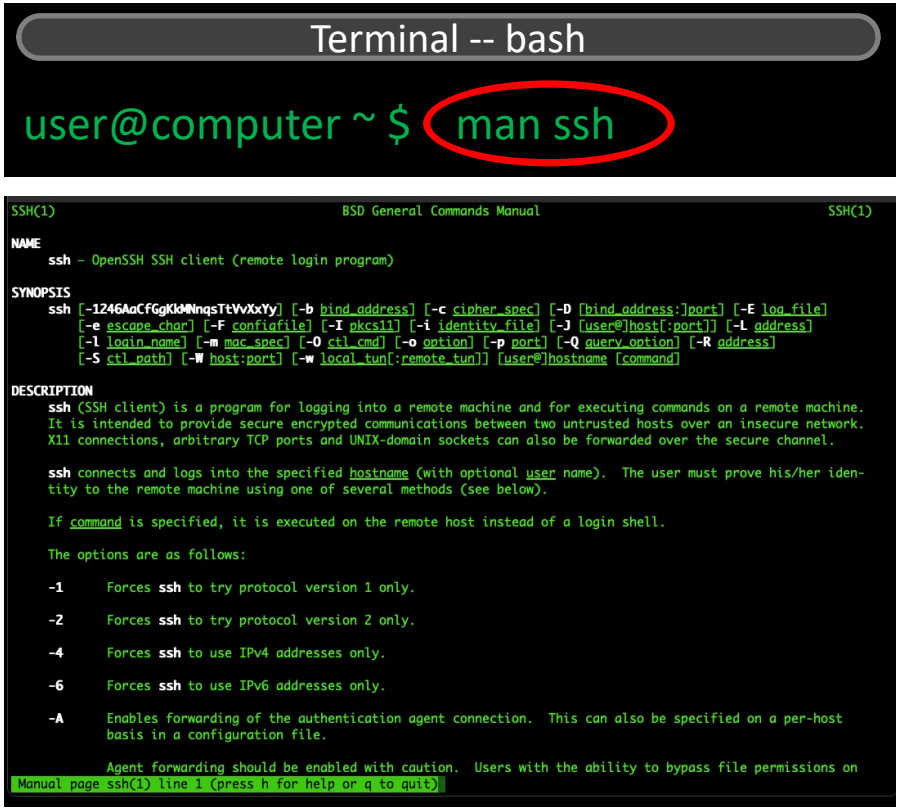


Linux

The Command Language

The system's **manual pager**, or **man page**, is the OS embedded reference manual

- Man pages divided into nine sections
- Section 1 contains the shell command pages
- Manual Command; **man** *command name* displays synopsis, description, and options

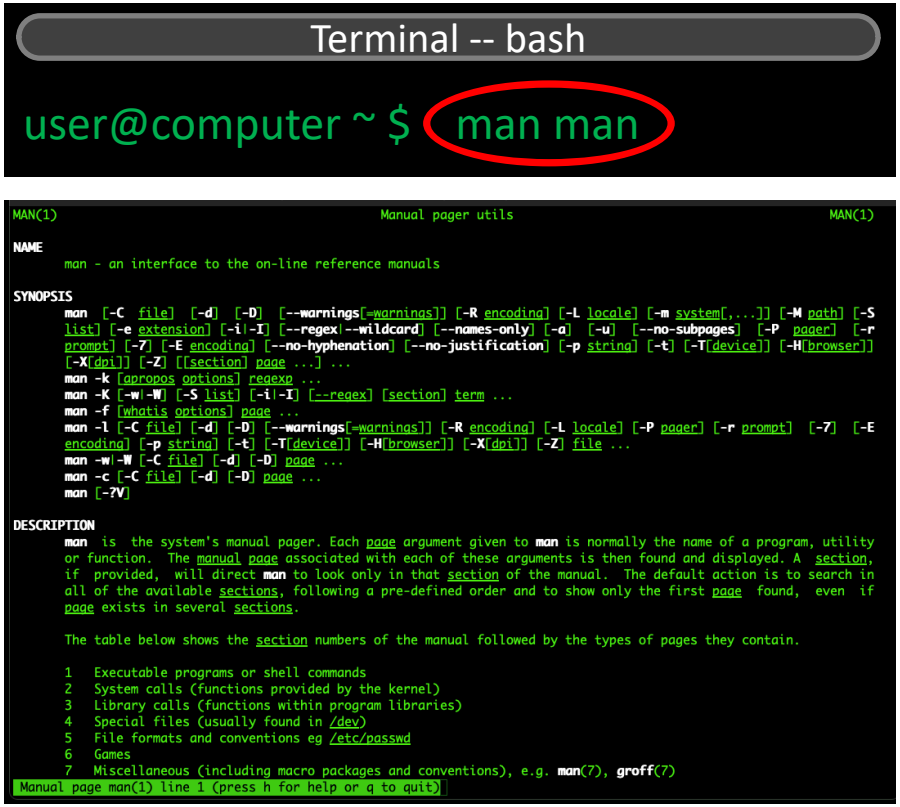


Linux

The Command Language

There is a **man page** for the **manual pager**

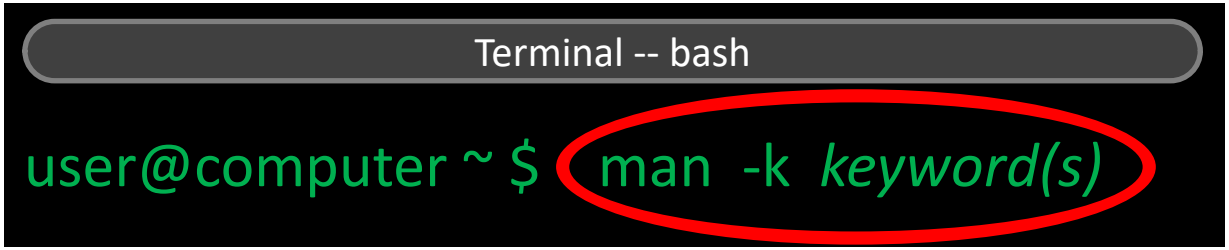
- Use **man** as both command and argument
- Details the sections, lists all man page options and provides some helpful examples



Linux

The Command Language

Search the **man pages** by keywords



- Command `man`
- Option `-k`
- Argument *Keyword(s)*

Linux

The Command Language

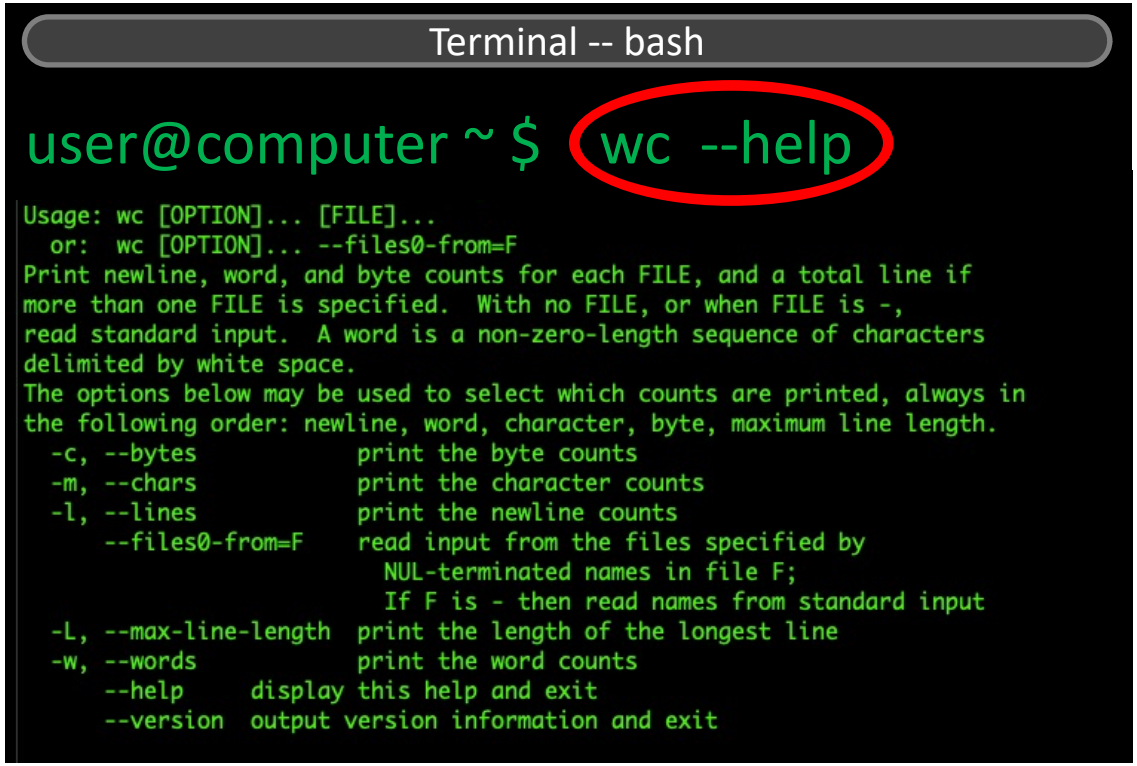
Example: Search the **man pages** for the command to login into the MSU HPCC

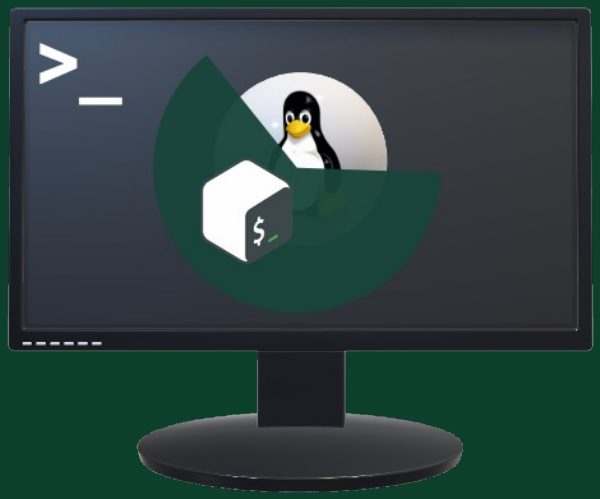
- Use **-s 1** to restrict search to Section 1 - Shell Commands

Linux

The Command Language

Use the **help** option (**-h** or **--help**) if you already know the command name





HPCC wiki

docs.icer.msu.edu

Contact ICER

icer.msu.edu/contact

ICER

INSTITUTE FOR CYBER-ENABLED RESEARCH

Linux

Linux on the HPCC

Mahmoud Parvizi, Ph.D.
Research Consultant, Institute for Cyber-Enabled Research
Research Associate, Department of Physics and Astronomy
parvizim@msu.edu

Linux

Linux on the HPCC

The MSU **High Performance Computing Center** (HPCC)

- **Clusters** of individual computers, or **nodes**, connected via a very fast **data interconnect**

- A **clustered filesystem**, a **module system** and a **job scheduler** manage data and perform tasks on the system

Linux

Linux on the HPCC

HPCC Hardware Definitions:

- **Cluster:** Set of computers that form a single system; comprises a network (~4 clusters)
- **Node:** Individual computer in a cluster; performs a task (~1,000 nodes)
- **Core:** Individual central processing unit, or CPU, in a node; performs computation (~60,000 CPUs, ~300 Tb memory)

Linux

Linux on the HPCC

HPCC Hardware Definitions:

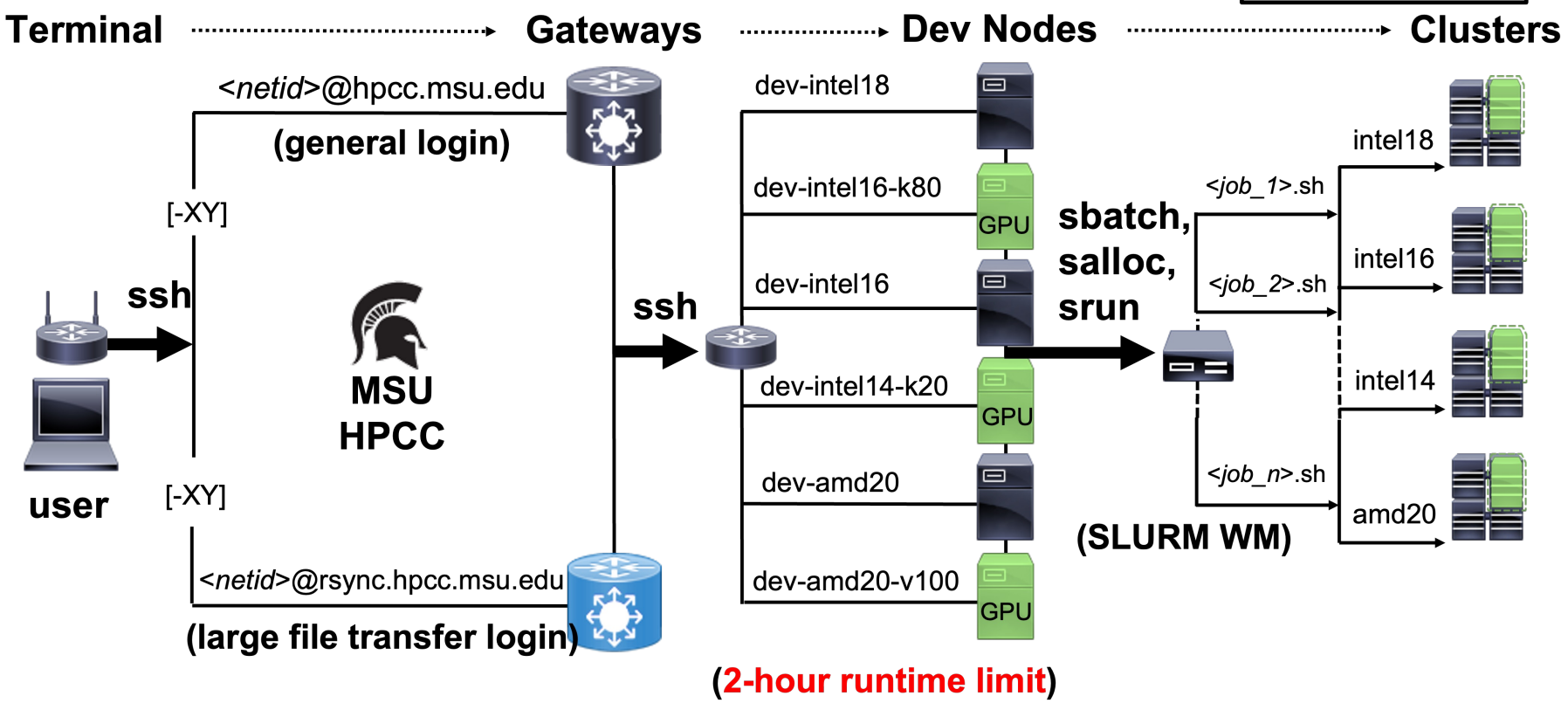
- **Data Interconnect:** Networking communications standard with high throughput and low latency

- **Infinband:** HPCC's data interconnect (~100Gb/s, ~8Pb storage)

Linux

Linux on the HPCC

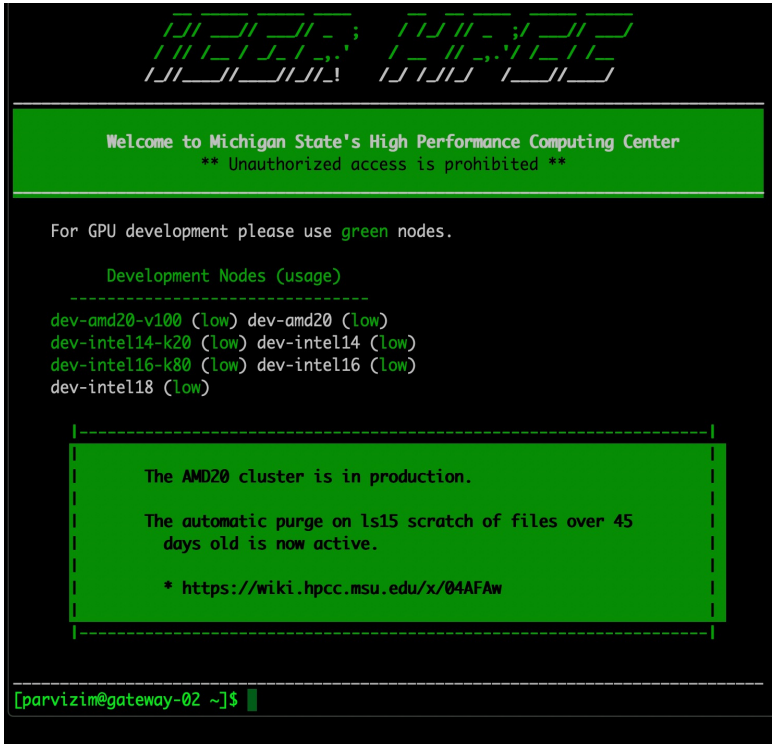
HPCC System Overview



Linux

Linux on the HPCC

Exercise: Login to the HPCC



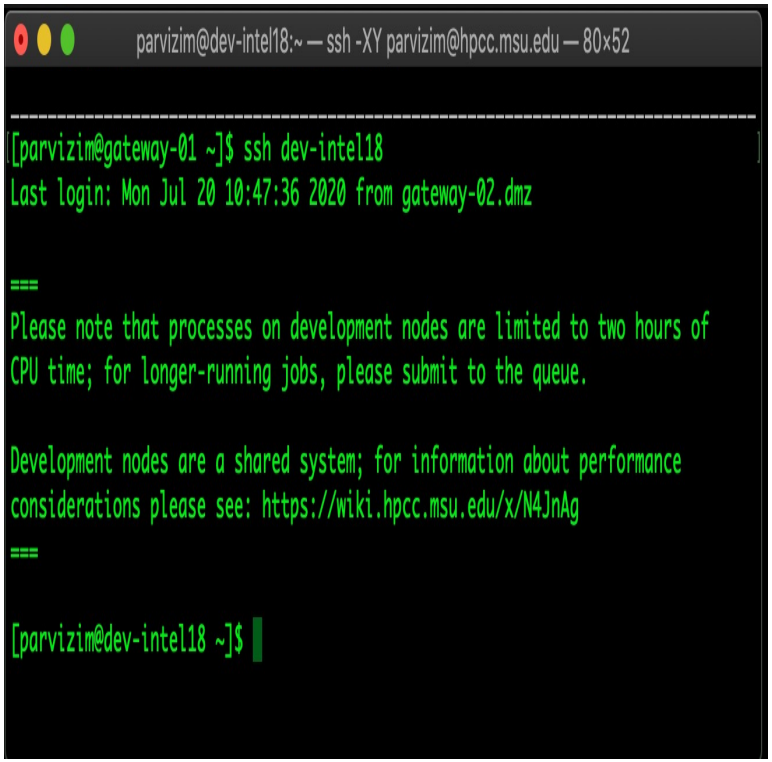
Type in Your Terminal:

[user@computer] $ ssh –X <*NetID*>@hpcc.msu.edu

Linux

Linux on the HPCC

Exercise: Login to dev-intel18



Type in Your Terminal:

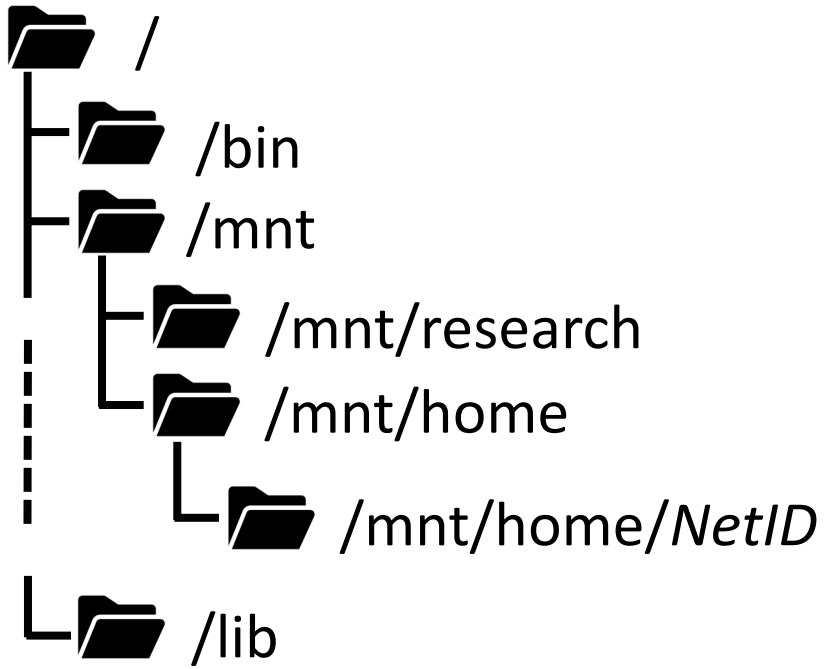
[user@computer] $ ssh -X dev-intel18

Linux

Linux on the HPCC

A **clustered filesystem** is a hierarchical collection of files accessible to all compute nodes of a cluster

- **File:** A formatted collection of bytes referenced by the OS
- **Directory:** Any file containing another file
- **Filesystem:** Method use by OS to store and retrieve files



Linux

Linux on the HPCC

Directories have a **tree-like** structure

- **Examples:**

/ -- the “root”

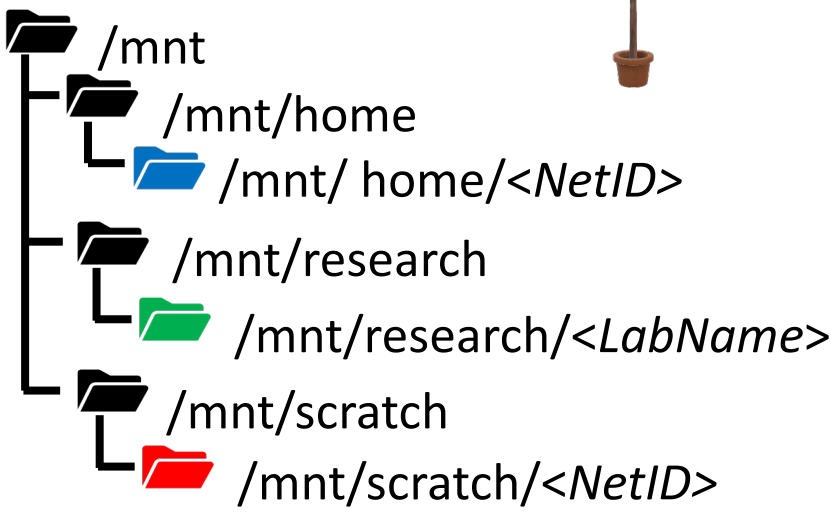
- **/bin** -- shared executables
- **/mnt** -- all groups and users
  - **/mnt/research** -- all research groups
  - **/mnt/home** -- all users
    - **/mnt/home/*NetID*** -- your personal directory
- **/lib** -- shared library of programs

Linux

Linux on the HPCC

User Spaces are directories users can write files to using any nodes in a cluster

- **Home:** Personal files and default login directory (50Gb storage)
- **Research:** Group research files (50Gb - 1Tb storage)
- **Scratch:** Temporary working files (~800 Tb total storage)



Linux

Linux on the HPCC

Exercise: use the **quota** command to display the details of your user spaces

Type in Your Terminal:

[user@computer] $ quota

```
parvizim@dev-intel18:~ — ssh parvizim@hpcc.msu.edu — 123×49

Development nodes are a shared system; for information about performance
considerations please see: https://wiki.hpcc.msu.edu/x/N4JnAg
===

[parvizim@dev-intel18 ~]$ quota
Home Directory:                 Space       Space       Space       Space       Files       Files       Files
Files
                                Quota       Used        Available   % Used      Quota       Used        Available
% Used
-------------------------------------------------------------------------------------------------------------------
------------

/mnt/home/parvizim              50G         7G          43G         14%         1048576     87299       961277
8%



Research Groups:                Space       Space       Space       Space       Files       Files       Files
Files
                                Quota       Used        Available   % Used      Quota       Used        Available
% Used
-------------------------------------------------------------------------------------------------------------------
------------

TOPMED                          4096G       3733G       363G        91%         4194304     1558        4192746
0%
UKBB                            9216G       8242G       974G        89%         9437184     6897        9430287
0%
helpdesk                        12288G      9888G       2400G       80%         52428800    44415439    8013361
85%



Temporary Filesystems:
-------------------------------------------------------------------------------------------------------------------
------------

/mnt/scratch (/mnt/gs18)        Space Quota Space Used  Space Free  Space % Used Filess Quota Files Used  Files Free
Files % Used
                                51200G      1G          51199G      0%          1048576     2           1048574
0%

/mnt/ls15 (legacy scratch)      Inodes Used   Quota       Free
                                1             1000000     999999

[parvizim@dev-intel18 ~]$
```

Linux

Linux on the HPCC

A **path to a file** is a list of the files containing the file of interest

Absolute **/mnt/home/*NetID*/data**

- /
- /mnt
- /mnt/home
- /mnt/home/*NetID*
- /mnt/home/*NetID*/data

Relative **. . /scripts**

- /mnt/home/*NetID*
- /mnt /home/*NetID*/data
- /mnt/home/*NetID*/scripts

Linux

Linux on the HPCC

Exercise: Find path with **pwd**, or print working directory, command

```
[parvizim@dev-intel18 ~]$
[parvizim@dev-intel18 ~]$
[parvizim@dev-intel18 ~]$ pwd
/mnt/home/parvizim
[parvizim@dev-intel18 ~]$
```

Type in Your Terminal:

[user@computer] $ pwd

Linux

Linux on the HPCC

Exercise: List files with **ls**, or list information, command

```
[parvizim@dev-intel18 ~]$
[parvizim@dev-intel18 ~]$ ls
Documents
[parvizim@dev-intel18 ~]$
```

Type in Your Terminal:

[user@computer] $ ls

Linux

Linux on the HPCC

Exercise: Enter ‘Documents’ with **cd**, or change directory, command and list the contents

```
[[parvizim@dev-intel18 ~]$
[[parvizim@dev-intel18 ~]$ cd Documents; ls
MATLAB  RT  Workshops
 [parvizim@dev-intel18 Documents]$
```

Type in Your Terminal:

[user@computer] $ cd Documents; ls

Linux

Linux on the HPCC

Exercise: Create a file ‘test.txt’ with the **cat**, or concatenate, command and the **>** operator

Type in Your Terminal:

[user@computer] $ cat > test.txt

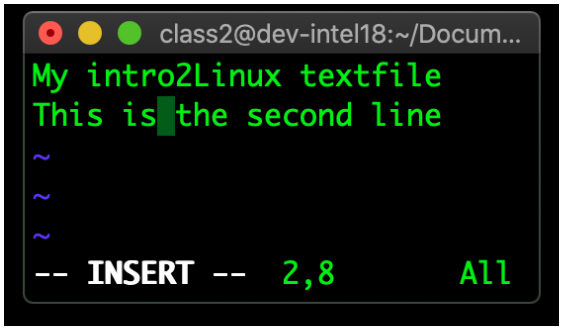
```
[parvizim@dev-intel18 Documents]$
[parvizim@dev-intel18 Documents]$
[parvizim@dev-intel18 Documents]$ cat > test.txt
My intro2linux textfile
the second line
^Z
[1]+  Stopped                 cat > test.txt
[parvizim@dev-intel18 Documents]$
```

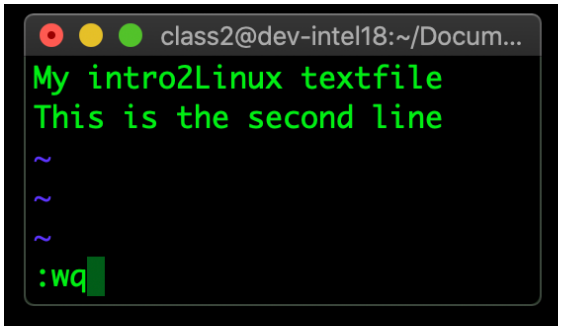
1) Type text: *My intro2Linux textfile* **[return]**
2) Type text: *the second line* **[return]**
3) Stop cat: **[control]+[Z]**

Linux

Linux on the HPCC

Exercise: Edit our text file with the **vi**, or vim editor, command





Type in Your Terminal:

[user@computer] $ vi test.txt

1) **[i],** look for the – INSERT – banner at bottom
2) Add text: "*This is*" to "*the second line*"
3) **[esc]**, – INSERT – banner gone
4) **[shift]+[:]** and look for **:** at bottom
5) Type *wq* to write and quit

Linux

Linux on the HPCC

Exercise: Change path back to ‘home’ and make a new directory with **mkdir**, or make directory, command

```
[parvizim@dev-intel18 Documents]$ cd ../; mkdir MyIntro2Linux; ls
MyIntro2Linux
[parvizim@dev-intel18 ~]$
```

Type in Your Terminal:

[user@computer] $ cd ../; mkdir MyIntro2Linux; ls

Linux

Linux on the HPCC

Exercise: Move the ‘test.txt’ file from ‘Documents’ to the new directory with **mv**, or move, command

```
[parvizim@dev-intel18 ~]$ mv ~/Documents/test.txt ~/MyIntro2Linux
[parvizim@dev-intel18 ~]$ cd MyIntro2Linux/; ls
test.txt
[parvizim@dev-intel18 MyIntro2Linux]$
```

Type in Your Terminal:

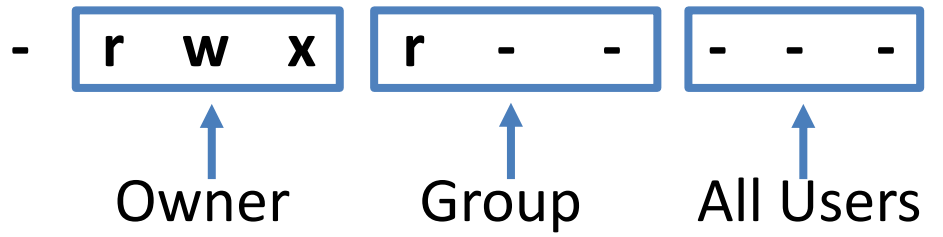
[user@computer] $ mv ~/Documents/test.txt ~/MyIntro2Linux

Linux

Linux on the HPCC

File Permissions determine which users may access, modify, and/or execute files

Permissions



r = Read
w = Write
x = Execute

Linux

Linux on the HPCC

chmod, or change mode, command is used to change file permissions

chmod **<user>** **<action>** **<permission>** **<FileName>**

u = Owner
g = Group
o = Others
a = All (u,g,o)

+ Give
- Take

r = Read
w = Write
x = Execute

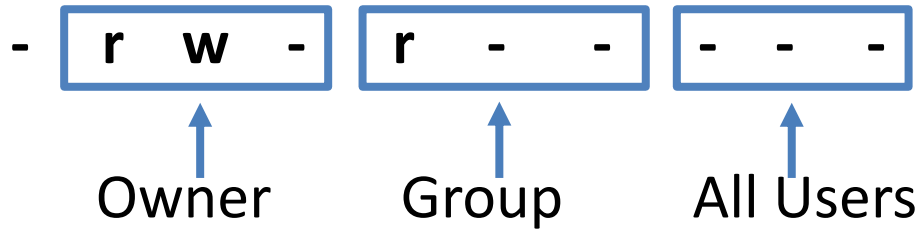



Linux

Linux on the HPCC

Exercise: list the permissions for ‘test.txt’ by using the **ls** command with the **– l,** or long, option

```
[parvizim@dev-intel18 MyIntro2Linux]$
[parvizim@dev-intel18 MyIntro2Linux]$ ls -l
total 1
-rw-r----- 1 parvizim helpdesk 40 Jan 24 01:10 test.txt
[parvizim@dev-intel18 MyIntro2Linux]$
```



Type in Your Terminal:

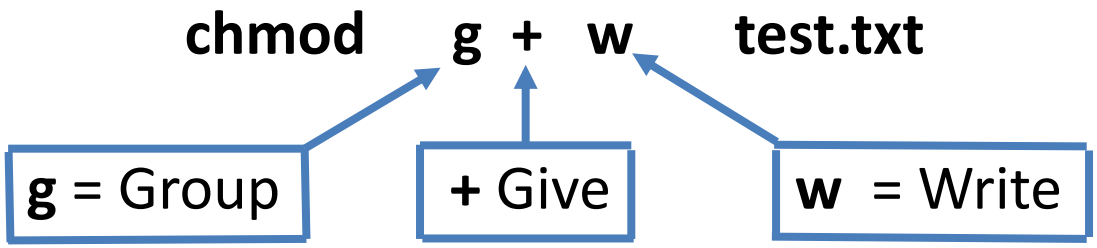
[user@computer] $ ls -l

Linux

Linux on the HPCC

Exercise: Change permissions of ‘test.txt’ to allow group members to write to the file

```
[parvizim@dev-intel18 MyIntro2Linux]$ chmod g+w test.txt; ls -l
total 1
-rw-rw---- 1 parvizim helpdesk 40 Jan 24 01:10 test.txt
[parvizim@dev-intel18 MyIntro2Linux]$
```



Type in Your Terminal:

[user@computer] $ cd MyIntro2Linux/; chmod g+w test.txt; ls -l

Linux

Linux on the HPCC

Exercise: Remove the ‘test.txt’ file from ‘MyIntro2Linux’ with **rm**, or remove, command

```
[[parvizim@dev-intel18 MyIntro2Linux]$
[[parvizim@dev-intel18 MyIntro2Linux]$ rm test.txt; ls
 [parvizim@dev-intel18 MyIntro2Linux]$
```

Type in Your Terminal:

[user@computer] $ rm test.txt; ls

Linux

Linux on the HPCC

Exercise: Remove the directory ‘MyIntro2Linux’ with **rmdir**, or remove directory, command

```
[[parvizim@dev-intel18 MyIntro2Linux]$
[[parvizim@dev-intel18 MyIntro2Linux]$ cd ~; rmdir MyIntro2Linux
[parvizim@dev-intel18 ~]$
```

Type in Your Terminal:

[user@computer] $ cd ~; rmdir MyIntro2Linux

Linux

Linux on the HPCC

Exercise: Open a 2nd terminal, create file ‘my.txt’, and copy it to the HPCC with **scp**, or secure copy, command

```
parvizim — -zsh — 106×24
Last login: Tue Jan 24 01:42:12 on ttys001
parvizim@ICERpro84 ~ % cat > my.txt
stuff
^Z
zsh: suspended  cat > my.txt
parvizim@ICERpro84 ~ % scp my.txt  parvizim@hpcc.msu.edu:~/Documents/my_copy.txt
Enter passphrase for key '/Users/parvizim/.ssh/id_rsa':
my.txt                                                100%    6     0.2KB/s   00:00
parvizim@ICERpro84 ~ %
```

Type in Your Terminal:

[user@computer] $ scp my.txt <NetID>@hpcc.msu.edu:~/Documents/my_copy.txt

Linux

Linux on the HPCC

Exercise: Copy file ‘my_copy.txt’, from the HPCC with **scp**, or secure copy, command

```
parvizim@ICERpro84 ~ %
parvizim@ICERpro84 ~ % scp parvizim@hpcc.msu.edu:~/Documents/my_copy.txt ./
Enter passphrase for key '/Users/parvizim/.ssh/id_rsa':
my_copy.txt                                   100%    6     0.1KB/s   00:00
parvizim@ICERpro84 ~ %
```

Type in Your Terminal:

[user@computer] $ scp <NetID>@hpcc.msu.edu:~/Documents/my_copy.txt ./

Linux

Linux on the HPCC

Exercise: Use **sfps**, or secure file transfer protocol, to move files to and from the HPCC

```
sftp> lpwd
Local working directory: /Users/mahmoudparvizi
sftp> pwd
Remote working directory: /mnt/ufs18/home-075/parvizim
sftp> cd Documents
sftp> pwd
Remote working directory: /mnt/ufs18/home-075/parvizim/Documents
sftp> get my_copy.txt
Fetching /mnt/ufs18/home-075/parvizim/Documents/my_copy.txt to my_copy.txt
sftp> put my.txt
Uploading my.txt to /mnt/ufs18/home-075/parvizim/Documents/my.txt
my.txt
sftp> quit
mahmoudparvizi@ShredBook-Pro ~ %
```

1. Use **lpwd** for local; **pwd** for HPCC
2. **cd** to your HPCC “Documents” directory
3. Use **get** my_copy.txt to copy *from* HPCC
4. Use **put** my.txt to copy *to* HPCC
5. Use **quit** to close sftp connection

Type in Your Terminal:

[user@computer] $ sftp <NetID>@hpcc.msu.edu

Linux

Linux on the HPCC

Exercise: Download ‘iris’ dataset from the web to HPCC with **wget**, or web get, command

```
[parvizim@dev-intel18 Documents]$ wget https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data
--2021-06-04 01:45:59--  https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data
Resolving archive.ics.uci.edu (archive.ics.uci.edu)... 128.195.10.252
Connecting to archive.ics.uci.edu (archive.ics.uci.edu)|128.195.10.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4551 (4.4K) [application/x-httpd-php]
Saving to: ‘iris.data’

100%[=====================================================================>] 4,551       --.-K/s   in 0s

2021-06-04 01:46:00 (41.1 MB/s) - ‘iris.data’ saved [4551/4551]

[parvizim@dev-intel18 Documents]$
```

Type in Your Terminal:

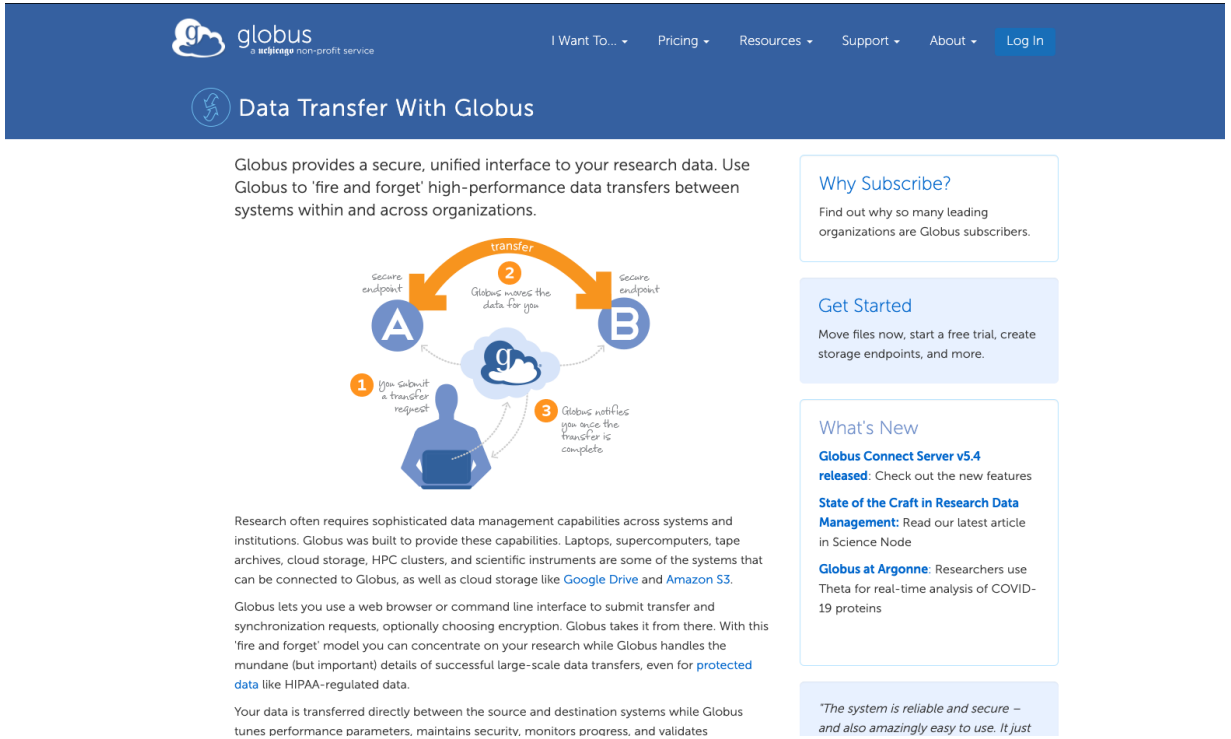
[user@computer] $ wget https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data

Linux

Linux on the HPCC

File transfer GUI **Globus** provided by MSU

https://www.globus.org/data-transfer



Linux

Linux on the HPCC

The HPCC **module system** manages the software environment you need to run your computations

- **Environment:** User specified software applications and their dependencies

- **Dependency:** Any file needed by an executable software application

- **Module:** User loaded file(s) that comprise an environment

Linux

Linux on the HPCC

Examples of HPCC modules include **compilers** and **libraries**

- **Compiler:** Software that translates code e.g., source to machine (GCC, intel, CUDA)

- **Library:** Collection of software resources used by the compiler and other executables; e.g., Math (BLAS, LaPACK)

Linux

Linux on the HPCC

Exercise: List default HPCC modules with the **module list** command

```
[parvizim@dev-intel18 ~]$
[parvizim@dev-intel18 ~]$ module list

Currently Loaded Modules:
  1) GCCcore/6.4.0      7) OpenBLAS/0.2.20                  13) CMake/3.11.1      19) libffi/3.2.1
  2) binutils/2.28      8) FFTW/3.3.7                       14) ncurses/6.0       20) Python/3.6.4
  3) GNU/6.4.0-2.28     9) ScaLAPACK/2.0.2-OpenBLAS-0.2.20  15) libreadline/7.0  21) Java/1.8.0_152
  4) OpenMPI/2.1.2     10) bzip2/1.0.6                      16) Tcl/8.6.8         22) MATLAB/2018a
  5) tbb/2018_U3       11) zlib/1.2.11                      17) SQLite/3.21.0
  6) imkl/2018.1.163   12) Boost/1.67.0                     18) GMP/6.1.2
```

Type in Your Terminal:

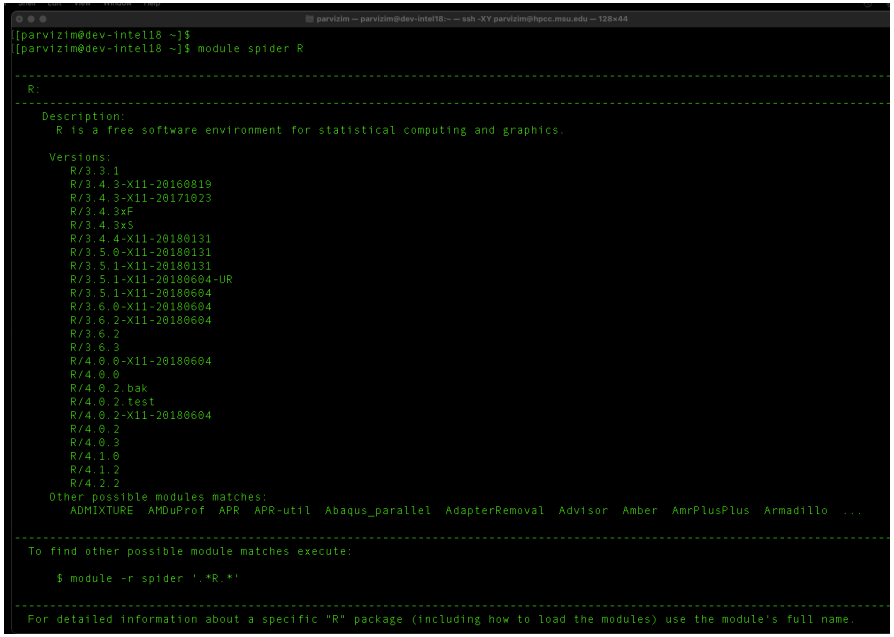
[user@computer] $ module list

Linux

Linux on the HPCC

Exercise: Find specific modules with the **module spider** command

Type in Your Terminal:



[user@computer] $ module spider R

Linux

Linux on the HPCC

Exercise: Find all HPCC modules with the **module avail** command

```
parvizim@dev-intel18:~ — ssh parvizim@hpcc.msu.edu — 123×49

---------------------------------- /opt/modules/MPI/GCC/6.4.0-2.28/OpenMPI/2.1.2 ----------------------------------
   ABySS/2.0.2                                        Pango/1.41.1
   ABySS/2.1.1                                        ParMETIS/4.0.3
   ABySS/2.1.5                              (D)       Perl/5.26.1
   ANTs/2.3.2                                         PyYAML/3.12-Python-2.7.14
   ATK/2.28.1                                         PyYAML/3.12-Python-3.6.4          (D)
   Armadillo/8.400.0                                  Python/2.7.14
   BBMap/37.93                                        Python/3.6.4                      (L)
   BCFtools/1.9                                       Qt5/5.10.1
   BLAST+/2.7.1                             (D)       QuantumESPRESSO/5.4.0-hybrid
   BUSCO/3.1.0-Python-3.6.4                           QuantumESPRESSO/6.2               (D)
   BWA/0.7.17                                         R/3.5.0-X11-20180131
   BamTools/2.5.1                                     R/3.5.1-X11-20180131              (D)
   BioPerl/1.7.2-Perl-5.26.1                          SAMtools/0.1.19
   Boost.Python/1.66.0-Python-3.6.4                   SAMtools/1.7
   Boost/1.66.0-Python-3.6.4                          SAMtools/1.9                      (D)
   Boost/1.66.0                                       SCOTCH/6.0.6
   Boost/1.67.0-Python-2.7.14                         SCons/3.0.1-Python-3.6.4
   Boost/1.67.0                             (L,D)     SDL2/2.0.9
   Bracken/2.2                                        SLiM/2019dev
   CGAL/4.11.1-Python-2.7.14                          SLiM/2021dev                      (D)
   CLASS/2.1.7                                        SOAPdenovo2/r241
   CONVERGE/2.4.21                                    SPAdes/3.11.1
   CONVERGE/2.4.27                                    SPAdes/3.13.0                     (D)
   CP2K/7.1                                           STAR/2.6.0c
   DMTCP/2.5.2                              (D)       SWIG/3.0.12-Python-3.6.4
   ELPA/2018.05.001                                   SageMath/8.8                      (D)
   ExaML/3.0.21                                       ScaLAPACK/2.0.2-OpenBLAS-0.2.20 (L)
   FFTW/3.3.7                               (L)       Stacks/2.0Beta10a
   FFmpeg/3.4.2                             (D)       Stacks/2.4                        (D)
   FLTK/1.3.4                                         Subread/1.6.2                     (D)
   FoX/4.1.2                                          SuiteSparse/5.1.2-METIS-5.1.0
   FriBidi/1.0.1                                      Tk/8.6.8                          (D)
   GATK/4.0.5.1-Python-3.6.4                          Tkinter/2.7.14-Python-2.7.14
   GATK/4.1.3.0-Python-3.6.4                          Tkinter/3.6.4-Python-3.6.4        (D)
   GATK/4.1.4.1-Python-3.6.4                (D)       Trilinos/12.12.1-Python-3.6.4
   GDAL/2.2.3-Python-3.6.4                            VCFtools/0.1.15-Perl-5.26.0
   GEOS/3.6.2-Python-3.6.4                            VCFtools/0.1.15-Perl-5.26.1       (D)
   GL2PS/1.4.0                                        VTK/7.1.1-Python-3.6.4
   GMAP-GSNAP/2018-05-11                              VTK/8.1.0-Python-3.6.4            (D)
   GObject-Introspection/1.54.1-Python-2.7.14         Valgrind/3.13.0
   GROMACS/2018                                       Velvet/1.2.10-mt-kmer_191
   GROMACS/201803                           (D)       Vim/8.2.0236-Python-3.6.4
   GStreamer/1.8.3                                    ZeroMQ/4.2.5
   GTK+/2.24.32                                       arpack-ng/3.5.0
   Gdk-Pixbuf/2.36.11                                 arpack-ng/3.6.3                   (D)
   GraphicsMagick/1.3.28                              awscli/1.16.109-Python-3.6.4
lines 1-48
```

Type in Your Terminal:

[user@computer] $ module avail

Linux

Linux on the HPCC

Exercise: Load HPCC modules with the **module load** command

```
[[parvizim@dev-intel18 ~]$ module load R/4.0.2
Lmod has detected the following error:  These module(s) or extension(s) exist but cannot be loaded as requested:
"R/4.0.2"
   Try: "module spider R/4.0.2" to see how to load the module(s).


[parvizim@dev-intel18 ~]$
```

Type in Your Terminal:

[user@computer] $ module load R/4.0.2

Linux

Linux on the HPCC

Exercise: Load HPCC modules with the **module load** command

```
[parvizim@dev-intel18 ~]$ module load GCC/8.3.0  OpenMPI/3.1.4 R/4.0.2

Lmod is automatically replacing "GNU/6.4.0-2.28" with "GCC/8.3.0".


Lmod is automatically replacing "GNU/6.4.0-2.28" with "GCC/8.3.0".

------------------------------------------------------------------------------------------------------------
------------------------
The following dependent module(s) are not currently loaded: OpenBLAS/0.2.20 (required by: ScaLAPACK/2.0.2
-OpenBLAS-0.2.20, Boost/1.67.0, Python/3.6.4, ScaLAPACK/2.0.2-OpenBLAS-0.2.20, Boost/1.67.0, Python/3.6.4
)
------------------------------------------------------------------------------------------------------------
------------------------

Inactive Modules:
  1) CMake/3.11.1     2) imkl/2018.1.163     3) tbb/2018_U3

Due to MODULEPATH changes, the following have been reloaded:
  1) Boost/1.67.0     2) GMP/6.1.2     3) libffi/3.2.1     4) zlib/1.2.11

The following have been reloaded with a version change:
  1) FFTW/3.3.7 => FFTW/3.3.8              8) ScaLAPACK/2.0.2-OpenBLAS-0.2.20 => ScaLAPACK/2.0.2
  2) GCCcore/6.4.0 => GCCcore/8.3.0        9) Tcl/8.6.8 => Tcl/8.6.9
  3) Java/1.8.0_152 => Java/11.0.2        10) binutils/2.28 => binutils/2.32
  4) OpenBLAS/0.2.20 => OpenBLAS/0.3.7    11) bzip2/1.0.6 => bzip2/1.0.8
  5) OpenMPI/2.1.2 => OpenMPI/3.1.4       12) libreadline/7.0 => libreadline/8.0
  6) Python/3.6.4 => Python/3.7.4         13) ncurses/6.0 => ncurses/6.1
  7) SQLite/3.21.0 => SQLite/3.29.0

[parvizim@dev-intel18 ~]$
```

Type in Your Terminal:

[user@computer] $ module load GCC/8.3.0 OpenMPI/3.1.4 R/4.0.2

Linux

Linux on the HPCC

Exercise: Unload HPCC modules with the **module unload** command

```
parvizim — parvizim@dev-intel18:~ — ssh -XY parvizim@hpcc.msu.edu — 102×27
Currently Loaded Modules:
  1) GCCcore/6.4.0      9) ScaLAPACK/2.0.2-OpenBLAS-0.2.20  17) SQLite/3.21.0
  2) binutils/2.28     10) bzip2/1.0.6                      18) GMP/6.1.2
  3) GNU/6.4.0-2.28    11) zlib/1.2.11                      19) libffi/3.2.1
  4) OpenMPI/2.1.2     12) Boost/1.67.0                     20) Python/3.6.4
  5) tbb/2018_U3       13) CMake/3.11.1                     21) Java/1.8.0_152
  6) imkl/2018.1.163   14) ncurses/6.0                      22) MATLAB/2018a
  7) OpenBLAS/0.2.20   15) libreadline/7.0                  23) powertools/1.2
  8) FFTW/3.3.7        16) Tcl/8.6.8

[parvizim@dev-intel18 ~]$ module unload powertools/1.2; module list

Currently Loaded Modules:
  1) GCCcore/6.4.0      9) ScaLAPACK/2.0.2-OpenBLAS-0.2.20  17) SQLite/3.21.0
  2) binutils/2.28     10) bzip2/1.0.6                      18) GMP/6.1.2
  3) GNU/6.4.0-2.28    11) zlib/1.2.11                      19) libffi/3.2.1
  4) OpenMPI/2.1.2     12) Boost/1.67.0                     20) Python/3.6.4
  5) tbb/2018_U3       13) CMake/3.11.1                     21) Java/1.8.0_152
  6) imkl/2018.1.163   14) ncurses/6.0                      22) MATLAB/2018a
  7) OpenBLAS/0.2.20   15) libreadline/7.0
  8) FFTW/3.3.7        16) Tcl/8.6.8

[parvizim@dev-intel18 ~]$
```

Type in Your Terminal:

[user@computer] $ module unload powertools/1.2; module list

Linux

Linux on the HPCC

Exercise: Unload all HPCC modules with **module purge** command

```
[parvizim@dev-intel18 ~]$
[parvizim@dev-intel18 ~]$ module list

Currently Loaded Modules:
  1) GCCcore/6.4.0      7) OpenBLAS/0.2.20                 13) CMake/3.11.1      19) libffi/3.2.1
  2) binutils/2.28      8) FFTW/3.3.7                      14) ncurses/6.0       20) Python/3.6.4
  3) GNU/6.4.0-2.28     9) ScaLAPACK/2.0.2-OpenBLAS-0.2.20 15) libreadline/7.0  21) Java/1.8.0_152
  4) OpenMPI/2.1.2     10) bzip2/1.0.6                     16) Tcl/8.6.8         22) MATLAB/2018a
  5) tbb/2018_U3       11) zlib/1.2.11                     17) SQLite/3.21.0
  6) imkl/2018.1.163   12) Boost/1.67.0                    18) GMP/6.1.2



[parvizim@dev-intel18 ~]$ module purge; module list
No modules loaded
[parvizim@dev-intel18 ~]$
```

Type in Your Terminal:

[user@computer] $ module purge; module list

Linux

Linux on the HPCC

Exercise: Reload default HPCC modules with the **logout** command

```
[parvizim@dev-intel18 ~]$
[parvizim@dev-intel18 ~]$ module purge; module list
No modules loaded
[parvizim@dev-intel18 ~]$ logout
Connection to dev-intel18 closed.

Currently Loaded Modules:
  1) gateway/1.0



[parvizim@gateway-03 ~]$ ssh dev-intel18
Last login: Mon Jan 23 21:39:23 2023 from gateway-03.dmz

===
Please note that processes on development nodes are limited to two hours of
CPU time; for longer-running jobs, please submit to the queue.

Development nodes are a shared system; for information about performance
considerations please see: https://docs.icer.msu.edu/development_nodes/
===

[parvizim@dev-intel18 ~]$ module list

Currently Loaded Modules:
  1) GCCcore/6.4.0      7) OpenBLAS/0.2.20                 13) CMake/3.11.1      19) libffi/3.2.1
  2) binutils/2.28      8) FFTW/3.3.7                      14) ncurses/6.0       20) Python/3.6.4
  3) GNU/6.4.0-2.28     9) ScaLAPACK/2.0.2-OpenBLAS-0.2.20 15) libreadline/7.0  21) Java/1.8.0_152
  4) OpenMPI/2.1.2     10) bzip2/1.0.6                     16) Tcl/8.6.8         22) MATLAB/2018a
  5) tbb/2018_U3       11) zlib/1.2.11                     17) SQLite/3.21.0     23) powertools/1.2
  6) imkl/2018.1.163   12) Boost/1.67.0                    18) GMP/6.1.2


[parvizim@dev-intel18 ~]$
```

Type in Your Terminal:

[user@computer] $ logout

Linux

Linux on the HPCC

Exercise: Write a ‘Hello World’ script with c

```c
#include <stdio.h>

int main()
{
        printf("Hello, World!\n");

        return 0;

}
```

Type in Your Terminal:

[user@computer] $ vi hello.c

1. #include <stdio.h>
2. int main()
3. {
4. printf(“Hello, World!\n”);
5. return 0;
6. }

Linux

Linux on the HPCC

Exercise: Load the GNU/8.2.0-2.31.1 compiler module

```
[parvizim@dev-intel18 ~]$
[parvizim@dev-intel18 ~]$ module purge; module load GNU/8.2.0-2.31.1
[parvizim@dev-intel18 ~]$
```

Type in Your Terminal:

[user@computer] $ module purge; module load GNU/8.2.0-2.31.1

Linux

Linux on the HPCC

Exercise: Compile and run the ‘Hello World’ script with GNU

```
[parvizim@dev-intel18 ~]$
[parvizim@dev-intel18 ~]$ gcc hello.c -o hello; ./hello
Hello, World!
[parvizim@dev-intel18 ~]$
```

Type in Your Terminal:

[user@computer] $ gcc hello.c -o hello; ./hello

Linux

Linux on the HPCC

The simple Linux utility for resource management (**SLURM**) is the HPCC workload manager i.e., **job scheduler**

- Framework for executing and monitoring jobs
- Allocates nodes/CPUs to users for a specified duration
- Manages the queue of pending jobs; arbitrates contentions

Linux

Linux on the HPCC

Exercise: Write a bash script ‘my_job.sb’ to schedule a SLURM job that runs your ‘hello.c’ script

```
#!/bin/bash                                  #Tell the shell to interpret bash

##### SLURM Resource Requests #####

#SBATCH --time=0-00:10                       #How long the job will run (days-hours:minutes)
#SBATCH --nodes=1                            #How many compute nodes the job needs
#SBATCH --ntasks=1                           #How many concurrent tasks the job needs
#SBATCH --cpus-per-task=1                    #How many CPUs each task needs
#SBATCH --mem-per-cpu=1G                     #How much memory each CPU needs

##### SLURM Administrative Settings #####

#SBATCH --job-name HelloWorld                #Name the job for convenience
#SBATCH --output=%x-%j.SLURMout              #Name the output file (JobName-JobNumber.SLURMout)
#SBATCH --mail-type=ALL                      #Tell SLURM to email you when job starts, stops, error
#SBATCH --mail-user=                         #Provide SLURM your email address


##### bash  Commands to Run #####

module purge                          #unload all modules

module load GNU/8.2.0-2.31.1          #load the GNU compiler

cd /mnt/home/                         #Navigate to the directory contaning hello.c

gcc hello.c -o hello                  #Run the command to compile hello.c

./hello                               #Run the compiled executable hello
```

Linux

Linux on the HPCC

Exercise: Submit ‘my_job.sb’ to SLURM with the **sbatch** command

```
[[parvizim@dev-intel18 ~]$ sbatch my_job.sb
Submitted batch job 3479290
[parvizim@dev-intel18 ~]$
```

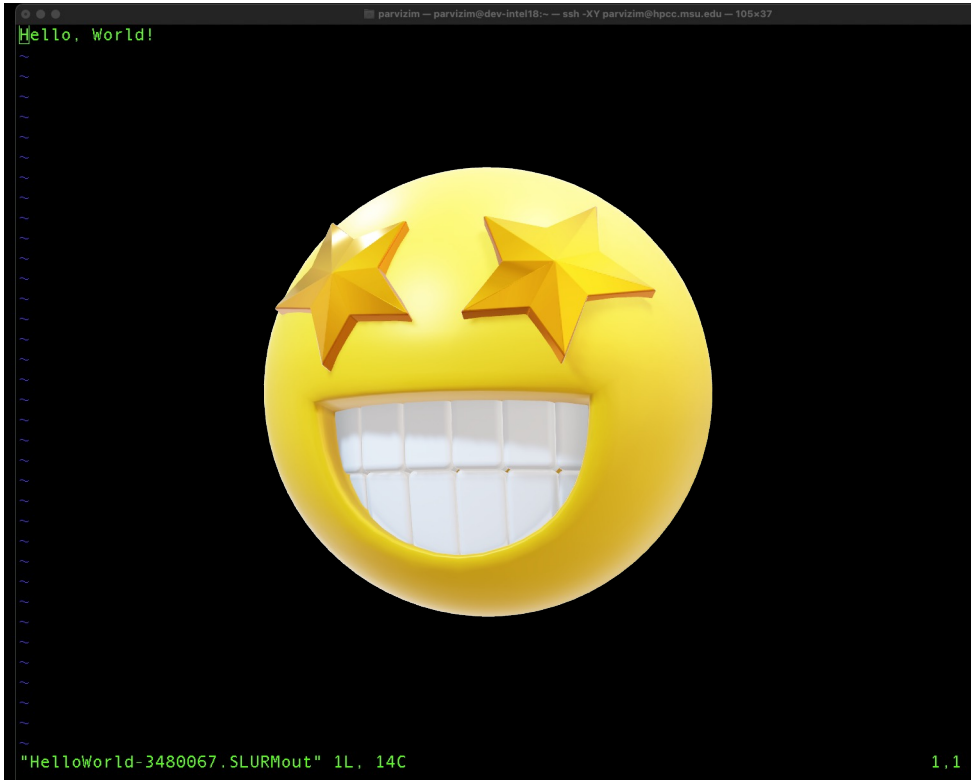
Type in Your Terminal:

[user@computer] $ sbatch my_job.sb

Linux

Linux on the HPCC

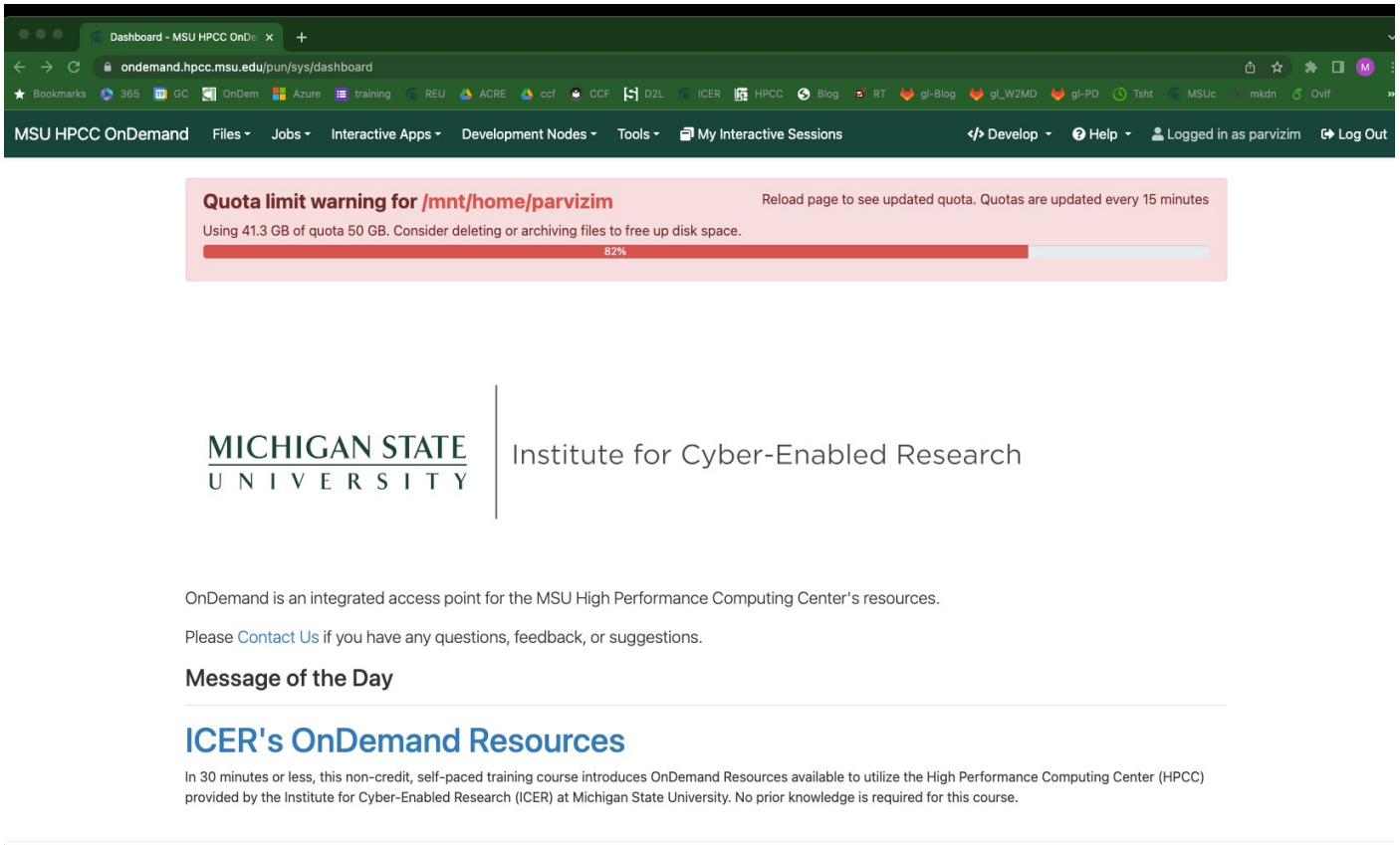
Exercise: Check the output file ‘HelloWorld-xxxxxxx.SLURMout’



Linux

Linux on the HPCC

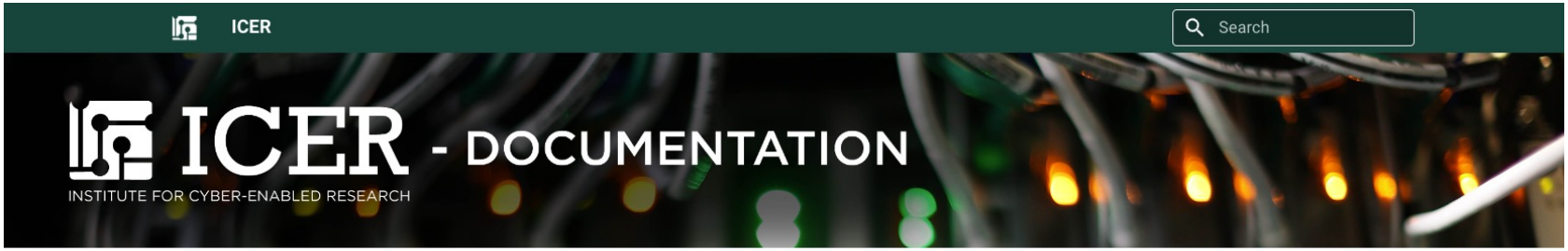
HPCC OnDemand ondemand.hpcc.msu.edu



Linux

Linux on the HPCC

HPCC Documentation, AKA “The Wiki”, docs.icer.msu.edu



MSU HPCC User Documentation

- Home
- Access the HPCC
- HPCC resources
- Computing environment
- File systems
- File transfer
- Software module system
- Powertools
- Job submission by SLURM
- Software tutorials
- Training material
- FAQ
- Lab Notebooks
- Tags

Home

Getting Access to the HPCC

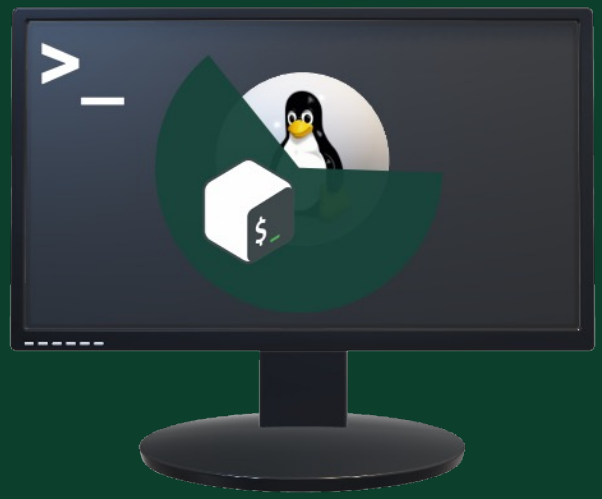
For potential users with an MSU NetID, accounts must be requested by a MSU tenure-track faculty member. Researchers at partner institutions should use the mechanism specified by their institution's agreement with MSU. For more information, see: Obtain an HPCC Account and on the ICER website.

CPU and GPU Time Limits

Non-buyin users are limited to 500,000 CPU hours (30,000,000 minutes) and 10,000 GPU hours (600,000 minutes) every year (from January 1st to December 31st). More information is available at Job Policies.

Table of contents

ICER
INSTITUTE FOR CYBER-ENABLED RESEARCH



Contact ICER

icer.msu.edu/contact