

# Lighthouse Cheat Sheet

Guide to general **Linux (Bash)** and **Slurm** commands

## Accessing Lighthouse

### Logging in from a terminal (Duo required)

```
ssh username@lighthouse.arc-ts.umich.edu
```

### Transferring files between Lighthouse and your system

```
scp source username@lighthouse-xfer.arc-ts.umich.edu:target  
scp -r source username@lighthouse-xfer.arc-ts.umich.edu:target  
scp username@lighthouse-xfer.arc-ts.umich.edu:source target
```

### GUI Clients

**PuTTY** SSH client for Windows

**WinSCP** SCP client for Windows

**FileZilla** FTP client for Windows, Mac, and Linux

## Basic Linux file management

man *command* Display the manual page for *command*

pwd Print out the present working directory

ls List the files in the current directory

ls -lh Show long, human-readable listing

ls *dir* List files inside directory *dir*

rm *file* Delete *file*

mkdir *dir* Create empty directory called *dir*

rmdir *dir* Remove empty directory *dir*

rm -r *dir* Remove directory *dir* and **all contents**

cd *dir* Change working directory to *dir*

cd .. Change working directory to parent

cd Change working directory to home

ls List the files in the current directory

cp *file1 file2* Copy *file1* as *file2*

cp *file1 dir* Copy *file1* into directory *dir*

mv *file1 file2* Rename *file1* as *file2*

mv *file1 dir* Move *file1* into directory *dir*

~ (tilde) Home directory

. (period) Current (working) directory

.. (2 periods) Parent directory

wget *URL* Download a file from Internet *URL*

unzip *file.zip* Extract a ZIP file

tar xzf *file* Extract a gzip compressed tarball (common extensions, *.tar.gz* and *.tgz*)

## Viewing and editing text files

cat *file* Print entire content of *file*

less *file* Similar to more, but with additional features

head *file* Print first 10 lines of *file*

tail *file* Print last 10 lines of *file*

nano Simple, easy to use text editor

vim Minimalist yet powerful text editor

emacs Extensible and customizable text editor

## Advanced file management

chmod Change read/write/execute permissions

which *cmd* List the full file path of a command

whereis *cmd* List all related file paths (binary, source, manual, etc.) of a command

du *dir* List size of directory and its subdirectories

find Find file in a directory

## Aliases and system variables

alias Create shortcut to command

env Lists all environment variables

export *var=val* Create environment variable *\$var* with value *val*

echo *\$var* Print the value of variable *\$var*

.bashrc File that defines user aliases and variables

## Input and output redirection

\$(*command*) Runs *command* first, then inserts output to the rest of the overall command

< Standard input redirection

> Standard output redirection

2> Standard error redirection

2>&1 Standard error to standard output redirection

*cmd1* | *cmd2* Pipe the output of *cmd1* to *cmd2*

## Filters

wc Word, line, and character count

grep Find and print text matching a regular expression

sort Sort input

uniq Filter duplicate lines

cut Cut specific fields or columns

sed Stream editor for search and replace

awk Extensive tool for complex filtering tasks

## Lighthouse directories

<code>/home/<i>username</i></code>	For use with running jobs, 80 GB quota
<code>/tmp</code>	Small file reads/writes, deleted after 10 days
<code>/scratch</code>	Large file reads/writes, purged periodically
<code>/afs</code>	Only on login node, 10 GB backed up

## Lmod

<code>module keyword <i>string</i></code>	Search for module names or descriptions matching <i>string</i>
<code>module spider <i>string</i></code>	Search for modules matching <i>string</i>
<code>module avail</code>	Show modules that can be loaded now
<code>module load <i>module</i></code>	Load <i>module</i> in the environment
<code>module show <i>module</i></code>	Show the help and variables set by <i>module</i>
<code>module list</code>	List currently loaded modules
<code>module unload <i>module</i></code>	Remove <i>module</i> from environment
<code>module purge</code>	Remove all modules from environment
<code>module save <i>collection</i></code>	Save all currently loaded modules to <i>collection</i>
<code>module savelist</code>	Return all saved module collections
<code>module describe <i>collection</i></code>	Return all modules in <i>collection</i>
<code>module restore <i>collection</i></code>	Restore all modules from <i>collection</i>

## Slurm

<code>sbatch <i>filename</i></code>	Submit a job script <i>filename</i>
<code>squeue -u <i>username</i></code>	Show job queue for user <i>username</i>
<code>scancel <i>jobid</i></code>	Delete job <i>jobid</i>
<code>scontrol hold <i>jobid</i></code>	Hold job <i>jobid</i>
<code>scontrol release <i>jobid</i></code>	Release job <i>jobid</i>
<code>sinfo</code>	Cluster status
<code>srun</code>	Launch parallel job step
<code>sacct</code>	Display job accounting information

## Slurm Environment Variables

<code>SLURM_JOBID</code>	Job ID
<code>SLURM_SUBMIT_DIR</code>	Job submission directory
<code>SLURM_SUBMIT_HOST</code>	Host from which job was submitted
<code>SLURM_JOB_NODELIST</code>	Node names allocated to job
<code>SLURM_ARRAY_TASK_ID</code>	Task ID within job array
<code>SLURM_JOB_PARTITION</code>	Job partition

## #SBATCH directives and #PBS counterparts

#SBATCH	#PBS	Description
<code>--job-name=<i>name</i></code>	<code>-N <i>name</i></code>	Job name
<code>--account=<i>name</i></code>	<code>-A <i>name</i></code>	Account to charge
<code>--partition=<i>name</i></code>	<code>-q <i>name</i></code>	Submit to partition (different for each PI)
<code>--time=<i>dd-hh:mm:ss</i></code>	<code>-l walltime=<i>time</i></code>	Time limit (walltime)
<code>--nodes=<i>count</i></code>	<code>-l nodes=<i>count</i></code>	Number of nodes
<code>--tasks-per-node=<i>count</i></code>	<code>-l ppn=<i>count</i></code>	Processes per node
<code>--cpus-per-task=<i>count</i></code>	<i>n/a</i>	CPU cores per process
<code>--mem=<i>count</i></code>	<code>-l mem=<i>count</i></code>	RAM per node (e.g. 1000M, 1G)
<code>--mem-per-cpu=<i>count</i></code>	<code>-l pmem=<i>count</i></code>	RAM per CPU core
<code>--gres=gpu:<i>count</i></code>	<code>-l gpus=<i>count</i></code>	GPUs per node
<code>--odelist=<i>nodes</i></code>	<code>-l nodes=<i>nodes</i></code>	Request nodes
<code>--array=<i>arrayspec</i></code>	<code>-t <i>arrayspec</i></code>	Define job array
<code>--output=%x-%j.<i>Log</i></code>	<code>-o <i>filepath</i></code>	Standard output in run directory, formatted: jobName-jobID.log
<code>--error=%x-%j-<i>E</i>.<i>Log</i></code>	<code>-e <i>filepath</i></code>	Standard error log
<code>--export=ALL</code>	<code>-V</code>	Copy environment
<code>--export=<i>var=val</i></code>	<code>-v <i>var=val</i></code>	Copy env variable
<code>--depend=<i>var:jobid</i></code>	<code>-W depend=<i>var:jobid</i></code>	Job dependency states ( <i>var</i> ): after, afterok, afterany, afternotok
<code>--mail-user=<i>email</i></code>	<code>-M <i>email</i></code>	Email for job alerts
<code>--mail-type=<i>type</i></code>	<code>-m <i>type</i></code>	Email alert types: BEGIN, END, NONE, FAIL, QUEUE
<code>--odelist=<i>nodes</i></code>	<i>n/a</i>	Preferred nodes
<code>--exclude=<i>nodes</i></code>	<i>n/a</i>	Nodes to avoid

## ARC-TS custom commands

<code>my_usage</code>	Usage in CPU minutes
<code>my_accounts</code>	Show account membership and resource limits
<code>slurm-quota -s</code>	Show user disk quota and usage

## ARC-TS Documentation & Support

Lighthouse User Guide:

<https://arc-ts.umich.edu/lighthouse/user-guide>

Email [hpc-support@umich.edu](mailto:hpc-support@umich.edu) for further Lighthouse support

Sensitive data should **not** be stored or processed on Lighthouse