

# Great Lakes Cheat Sheet

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

## Accessing Great Lakes

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

ssh **username@greatlakes.arc-ts.umich.edu**

### Transferring files between Great Lakes and your system

scp **input username@greatlakes-xfer.arc-ts.umich.edu:output**

scp -r **input username@greatlakes-xfer.arc-ts.umich.edu:output**

scp **username@greatlakes-xfer.arc-ts.umich.edu:input output**

### 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** Prints content of file page by page

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

## Great Lakes directories

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

## Lmod

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

## Slurm

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

## Slurm Environment Variables

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

## #SBATCH directives and #PBS counterparts

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

## ARC-TS custom commands

my_usage	Usage in CPU minutes
my_accounts	Show account membership and resource limits
home-quota -s	Show user disk quota and usage
maxwalltime	Show walltime available for jobs (including upcoming maintenance)

## ARC-TS Documentation & Support

Great Lakes User Guide:

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

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

Sensitive data should **not** be stored or processed on Great Lakes