

Great Lakes Cheat Sheet

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

Basic Linux file management

| | | |
|----------------------|--------------------------|--|
| <code>man</code> | <code>command</code> | Display the manual page for <code>command</code> |
| <code>pwd</code> | | Print out the present working directory |
| <code>ls</code> | | List the files in the current directory |
| <code>ls -lh</code> | | Show long, human-readable listing |
| <code>ls</code> | <code>dir</code> | List files inside directory <code>dir</code> |
| <code>rm</code> | <code>file</code> | Delete <code>file</code> |
| <code>mkdir</code> | <code>dir</code> | Create empty directory called <code>dir</code> |
| <code>rmdir</code> | <code>dir</code> | Remove empty directory <code>dir</code> |
| <code>rm -r</code> | <code>dir</code> | Remove directory <code>dir</code> and all contents |
| <code>cd</code> | <code>dir</code> | Change working directory to <code>dir</code> |
| <code>cd ..</code> | | Change working directory to parent |
| <code>cd</code> | | Change working directory to home |
| <code>ls</code> | | List the files in the current directory |
| <code>cp</code> | <code>file1 file2</code> | Copy <code>file1</code> as <code>file2</code> |
| <code>cp</code> | <code>file1 dir</code> | Copy <code>file1</code> into directory <code>dir</code> |
| <code>mv</code> | <code>file1 file2</code> | Rename <code>file1</code> as <code>file2</code> |
| <code>mv</code> | <code>file1 dir</code> | Move <code>file1</code> into directory <code>dir</code> |
| <code>~</code> | (tilde) | Home directory |
| <code>.</code> | (period) | Current (working) directory |
| <code>..</code> | (2 periods) | Parent directory |
| <code>wget</code> | <code>URL</code> | Download a file from Internet <code>URL</code> |
| <code>unzip</code> | <code>file.zip</code> | Extract a ZIP file |
| <code>tar xzf</code> | <code>file</code> | Extract a gzip compressed tarball (common extensions: <code>.tar.gz</code> and <code>.tgz</code>) |

Viewing and editing text files

| | | |
|--------------------|-------------------|---|
| <code>cat</code> | <code>file</code> | Print entire content of <code>file</code> |
| <code>less</code> | <code>file</code> | Prints content of file page by page |
| <code>head</code> | <code>file</code> | Print first 10 lines of <code>file</code> |
| <code>tail</code> | <code>file</code> | Print last 10 lines of <code>file</code> |
| <code>nano</code> | | Simple, easy to use text editor |
| <code>vim</code> | | Minimalist yet powerful text editor |
| <code>emacs</code> | | Extensible and customizable text editor |

Advanced file management

| | | |
|----------------------|------------------|---|
| <code>chmod</code> | | Change read/write/execute permissions |
| <code>which</code> | <code>cmd</code> | List the full file path of a command |
| <code>whereis</code> | <code>cmd</code> | List all related file paths (binary, source, manual, etc.) of a command |
| <code>du</code> | <code>dir</code> | List size of directory and its subdirectories |
| <code>find</code> | | Find file in a directory |

Aliases and system variables

| | | |
|-----------------------|----------------------|--|
| <code>alias</code> | | Create shortcut to command |
| <code>env</code> | | Lists all environment variables |
| <code>export</code> | <code>var=val</code> | Create environment variable <code>\$var</code> with value <code>val</code> |
| <code>echo</code> | <code>\$var</code> | Print the value of variable <code>\$var</code> |
| <code>. bashrc</code> | | File that defines user aliases and variables |

Input and output redirection

| | | |
|--------------------------|--|---|
| <code>\$(command)</code> | | Runs <code>command</code> first, then inserts output to the rest of the overall command |
| <code><</code> | | Standard input redirection |
| <code>></code> | | Standard output redirection |
| <code>2></code> | | Standard error redirection |
| <code>2>&1</code> | | Standard error to standard output redirection |
| <code>cmd1 cmd2</code> | | Pipe the output of <code>cmd1</code> to <code>cmd2</code> |

Filters

| | | |
|-------------------|--|---|
| <code>wc</code> | | Word, line, and character count |
| <code>grep</code> | | Find and print text matching a regular expression |
| <code>sort</code> | | Sort input |
| <code>uniq</code> | | Filter duplicate lines |
| <code>cut</code> | | Cut specific fields or columns |
| <code>sed</code> | | Stream editor for search and replace |
| <code>awk</code> | | 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>user</i> OR sq <i>user</i> | Show job queue for <i>user</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 info |

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 |
| --odelist= <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 |