



University of  
**BRISTOL**

## BlueCrystal Reference Card

Advanced Computer Research Centre  
<https://www.acrc.bris.ac.uk/>

### Logging In

**ssh** Secure shell.  
options include:  
**-X** Enables X11 forwarding.  
example:  
**ssh user@bluecrystalp3**

### Transferring Files

**scp** Secure copy.  
options include:  
**-r** Recursively copy entire directories.  
examples:  
**scp local.tar.gz user@bluecrystalp2:~**  
**scp -r user@bluecrystalp2:/remote/path .**

**winscp** scp/sftp GUI for windows.

### Checking Your Quota

BCp2: **showquota**, BCp3: **pan\_quota**

### Configuring Your Environment

**module** Interface to modules package.  
options include:  
**avail** List all available modulefiles.  
**add** Load modulefile into shell environment.  
**del** Remove modulefile from shell environment.  
**list** List loaded modulefiles.

**whatis** Display the modulefile information.  
example:

**module add openmpi/gcc/64/1.6.5**

**\$HOME/.bashrc** BASH startup file

Make your environment customisation persistent, and available on compute nodes, by storing, e.g. your module commands in this file.

### Using the Queuing System

**qsub** Submit job to queuing system.  
example:

**qsub submission-script**

where the following submission script contains the minimal information required to submission a serial program to the *veryshort* queue (wallclock limit 2hrs):

```
#!/bin/bash
#PBS -l nodes=1:ppn=1,walltime=02:00:00
cd $PBS_O_WORKDIR
./my-serial-program
```

for a multi-threaded program:

```
#!/bin/bash
#PBS -l nodes=1:ppn=4,walltime=02:00:00
cd $PBS_O_WORKDIR
./my-multi-threaded-program
```

for an MPI (distributed memory parallel) program:

```
#!/bin/bash
#PBS -l nodes=2:ppn=4,walltime=02:00:00
cd $PBS_O_WORKDIR
cat $PBS_NODEFILE > machine.file.$PBS_JOBID
numnodes=`wc $PBS_NODEFILE | awk '{print $1}'`
mpirun -np $numnodes -machinefile \
machine.file.$PBS_JOBID ./my-mpi-program
```

Ensure that you have the appropriate module add commands in your shell start-up file (e.g. ~/.bashrc).

**qstat** Show status of batch jobs.  
options include:

**-u user** Show only user's jobs.

**-n** List nodes allocated to a job.

**showq** Show status of batch jobs—alternative format.

**qdel** Delete batch job with given job ID.

example:

**qdel 673615**

### Some Useful Linux Commands

#### File/Directory Basics

**ls** List directory contents.

examples:

**ls -ltr** Long listing, most recently modified last.

**ls -lh** File sizes in readable format e.g. 1K, 234M.

**pwd** Print working directory.

**cd** Change current directory.

examples:

**cd ..** Change to directory above.

**cd /path/to/dir** Change to directory given in path.

**cp** Copy files and directories.

examples:

**cp file1 file2** create a copy of file1 called file2.

**cp -r dir1 dir2** recursively copy dir1.

**mv** Move (rename) files and directories.

examples:

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

**mv dir1 /new/path** Move dir1 to a new location.

**rm** Remove files or directories.

examples:

**rm -i file1** Prompt before deleting file1.

**rm -rf dir1** Recursively & forcefully remove dir1.

**mkdir** Make directories.

**rmdir** Remove empty directories.

**ln** Make links between files and directories.

examples:

**ln -s /path/to/dir1 ./dir1** Symbolically link to dir1.

Viewing & Manipulating Text Files		example:		example:
<b>head</b> Output the first part of files.		<b>awk '{print \$2}' file1.txt</b>	Print second column of file1.txt.	<b>du -sh .</b> Summarize (in readable format) total usage of file-tree rooted in current dir.
example:		<b>gedit</b> Text editor.		<b>df</b> Report file system disk space usage.
<b>head -7 file.txt</b> View first 7 lines of file.txt.		<b>emacs</b> Text editor.		example:
<b>tail</b> Output the last part of files.		<b>Redirection and Pipelines</b>		<b>df -h .</b> Report usage (including available space) for file system holding current dir.
example:		> Redirect stdout.		<b>Searching For Things</b>
<b>tail -7 file.txt</b> View last 7 lines of file.txt.		example:		<b>grep</b> Print lines matching a pattern.
<b>cat</b> Concatenate files and print to stdout.		<b>cat file1 file2 &gt; file1-and-2</b>		examples:
example:		< Redirect stdin.		<b>grep -n 'foo' file.txt</b> Print all lines (prefixing the line number) containing 'foo' in file.txt.
<b>cat file1.txt file2.txt</b> Append file2.txt to file1.txt.		>> Redirect stdout and append.		<b>grep -i 'foo' *</b> Print all lines containing 'foo' (case insensitive) from all files in current dir.
<b>wc file.txt</b> Print line, word and byte counts.		example		<b>find</b> Search for files in a directory hierarchy.
<b>diff file1.txt file2.txt</b> Compare files, line by line.		<b>cat file1-and-2 file3 &gt;&gt; file-1-and-2-and-3</b>		<b>which</b> Locate a command.
<b>cut</b> Print selected parts from each line of files.		Pipe stdout from one cmd to stdin of another.		example:
example:		example:		<b>which gcc</b> Report location of gcc compiler.
<b>cut -d',' -f1,2 file.csv</b> Print first two columns of file.csv.		<b>head -7 file1   tail -1</b> view 7th line of file1		<b>whoami</b> Print effective userid.
<b>paste</b> Merge lines of files.		<b>Viewing Other Files</b>		<b>man</b> An interface on on-line reference manuals.
example:		<b>od</b> Dump files in octal and other (e.g. binary) formats.		<b>info</b> Read Info documents.
<b>paste file1.txt file2.txt</b> Concatenate each line of file1.txt and file2.txt, in turn, and print.		<b>nm</b> List symbols from object (& library) files.		? Wildcard: matches a single character.
<b>sort</b> Sort lines of files.		example:		* Wildcard: matches any sequence of characters.
examples:		<b>nm mylib.a   less</b> view symbols in mylib.a, one page at a time.		<b>Compressing and Combining</b>
<b>sort -d file1.txt</b> Print contents of file1.txt in dictionary order.		<b>ldd</b> Report shared library dependencies.		<b>tar</b> Archiving utility.
<b>sort -nr file1.txt</b> Print contents in reversed (descending) numerical order.		example:		example:
<b>uniq</b> Report or omit repeated lines.		<b>ldd myprog.exe</b> View myprog.exe's dependencies.		<b>tar -xzf archive.tar.gz</b> Unpack compressed archive.
example:		<b>File Properties</b>		<b>gzip</b> Compress files.
<b>uniq file1.txt</b> Print only unique lines of file1.txt.		<b>file</b> Determine file type.		example:
<b>sed</b> Stream editor for filtering and transforming.		<b>touch</b> Change file timestamps.		<b>gzip file.txt</b> Compress file.txt.
examples:		example:		<b>gunzip</b> Expand files.
<b>sed 's/cat/bat/g' file1.txt</b> replace all instances of 'cat' in file1.txt with 'bat'.		<b>touch file1</b> Updates access and modification times of file1 to the present time.		<b>Process Management</b>
<b>sed 's/*ed//g' file1.txt</b> replace all words in file1.txt ending with 'ed' with the empty string.		<b>chmod</b> Change file mode bits.		<b>top</b> Display Linux tasks.
<b>awk</b> Pattern scanning and processing language.		example:		<b>kill</b> Send a signal to a process.
		<b>chmod a+r file.txt</b> Allow all to read file.txt.		<b>fg</b> Place a job in the foreground.
		<b>chown</b> Change file owner and group.		<b>bg</b> Place a job in the background.
		<b>md5sum</b> Compute/check MD5 message digest.		
		<b>du</b> Estimate file space usage.		