Basic Commands		
pwd		Print working directory
cd	cd ~/directory/	Change Directory
mkdir	mkdir newdirectory	Make Directory newdirectory
mv	mv [-f] ~/source ~/destination	Move or rename file or directory
cp	cp [-fr] ~/source ~/newcopy	Copy a file or directory
ls	ls [-lhtr] [file][directory]	list all files or files in directory
rm	rm ~/file	Delete file file
man	man command	Show manual page on command command
ssh	ssh hostname [-l username]	Connect to remote host with user name username
scp	scp ~/source hostname:~/dest/	Copy file source to host hostname
vi	vi [~/file]	A powerful text editor
nano	nano [~/file]	A pico like text editor
emacs	emacs [~/file]	A powerful text editor
grep	grep [-iv] string file	Search for string in file
wc	wc [-l] file	Count the number of word/lines/chars in file
tail	tail [-f][-n#] file	Show last few lines of file
head	head [-n#] file	Show first few lines of file
dos2unix	dos2unix file	Change dos files into Unix/Linux files.
df	df [-h][/home]	Show the size and amount free of your quota
du	du [-sh][directory]	Show total size of items in directory

PBS Commands		
qstat	qstat [-u user][-a][-n][jobid]	Show current state of queues
qsub	qsub [-l resources][-I] pbsfile	Submit job with definitions in pbsfile
qdel	qdel jobid	Delete queued/running job with id jobid
qalter	qalter [-l walltime=#] jobid	Alter required walltime/memory for job jobid
checkjob	checkjob jobid	Verbose information on job jobid
qpeek	qpeek [-fce –help] jobid	View STDOUT and STDERR of running job

Module Commands		
list	module list	Show current loaded modules
load	module load software[/version]	Load software module and version
rm	module rm software	Removed loaded software software
swap	module swap old new[/version]	Swap module org for module new/version
avail	module avail [software]	Show all available modules
show	module show software[/version]	Show what module software does

Software Development		
`mpicc`	`mpicc [options] source.c`	Compile MPI C source
`mpiCC`	`mpiCC -fPIC [options] source.cpp`	Compile MPI C++ source
`mpif90`	`mpif90 [options] source.f90`	Compile MPI F90/F77 source
`mpirun`	`mpirun -np # executable`	Run `executable` on # cpus.
`pgcc`	`pgcc [options] source.c`	Recommended C compiler
`pgCC`	`pgCC -fPIC [options] source.cpp`	Recommended C++ compiler
`pgf90`	`pgf90 [options] source.f90`	Recommended F90/F77 compiler
`time`	`time expression`	Time how long `expression` takes
`make`	`make [-j#][-f makefile]`	Evaluate `makefile` in current directory
`ddt`	`ddt executable`	Parallel graphical debugger
`opt-gui`	`opt-gui`	Parallel Profiler
`diff`	`diff file1 file2`	Show changes between two files

Tips

- DON'T run software on the login node, many people use it at once
- DO use debug nodes. There are machines for jobs 15 minutes or less (`#PBS -l walltime=15:00`) to test pbs scripts and debug software. You can also use interactive jobs to access them:
  (`qsub -I -l walltime=15:00 pbsscript`)
- DO use compiler optimizations.
- DO use at least `-fast` with PGI compilers and `-fastsse` if possible.
- DO always verify results when changing compiler/optimizations/architecture or math libraries.
- DON'T write your own matrix math and linear algebra functions
- DO use the BLAS (www.netlib.org/blas) and LAPACK (www.netlib.org/lapack)libraries.
- DO use vendor optimized BLAS libraries when able, ACML (http://developer.amd.com/acml.jsp) on `nyx.engin`
- DO use `RAND_MAX` as the maximum value returned by `rand() (man 3 rand)`
- DON'T use `rand()` if you need more than $(2^{32}/2)-1$ values.
- DO look at SPRNG (http://sprng.cs.fsu.edu/) and ACML for faster PRNG's with longer periods.
- DO communicate with other users in your group effectively on using CAC resources
- DO communicate with the CAC at `cac-support@umich.edu`