

## UNIX Commands

ls – list contents of a directory  
pwd – display current directory  
cd – change to directory (cd test)  
cp – copy file (cp from\_file to\_file)  
mv – move file (mv from\_file to\_file)  
rm – delete file (rm test1)  
mkdir – make directory (mkdir test)  
rmdir – remove directory (rmdir test)  
man – help for given command (man cp)  
cat – display contents of file  
more – display contents of file with page breaks (next page with Space key)  
head – display top of file  
tail – display end of file  
grep – search for pattern in file (grep "pattern" test1)  
vi – edit file (more on this later)  
wc – count number of lines, words, characters in a file (wc -l file.txt)  
cut – cut out parts of file (cut -f 2 -d : file.txt)  
du – report disk usage (du -h)  
df – report file systems and their usage (df -h)  
chmod – change permissions of a file (chmod a+r file.txt)  
ln – create a link to a file (ln -s ~/bin/prog.exe prog1.exe)

### Wildcards:

\* - matches any number of letters incl. none  
? - matches any single character  
[] - encloses set of characters that can match the single given position  
- used within [] denotes range of characters  
~ - followed by user name = home directory (~mcuma)

### Command redirection:

> redirect to a new file (cat test1 > test3)  
>> - append to a (new or existing) file (cat test2 >> test3)  
| - pipe – redirect command output to another command

## Vi basic reference

i – insert  
I – insert at start of line  
a – append  
A – append at end of line  
o – enter insert mode on a new line  
r – replace single letter  
R – replace mode  
x – delete letter  
dd – delete line  
D – delete from cursor to end of line  
u – undo

G – move to end of file  
#G – move to line #  
^ - move to start of line  
\$ - move to end of line  
ctrl-f – move page forward  
ctrl-b – move page back

:w – save current file  
:w filename – save file to new file  
:q – quit vi  
:wq – save and quit  
:q! - quit without saving  
:w! - force save  
:n filename – open another file

/pattern – search forward for pattern  
? - search backward for pattern; subsequent entry of / or ? searches for next occurrence of pattern

:s/old\_text/new\_text – replace next occurrence on current line  
:s/old\_text/new\_text/g – replace all occurrence on current line  
:%s/old\_text/new\_text/g – replace all occurrences in the whole file