## C shell scripts

mylatex1.csh

```
#!/bin/csh

latex file.tex
dvips -f file.dvi > file.ps
rm file.aux file.dvi file.log
```

mylatex2.csh

```
#!/bin/csh

latex $1.tex
dvips -f $1.dvi > $1.ps
rm $1.aux $1.dvi $1.log
```

## Command substitution in a script

To include the output from one command within the command line for another command, enclose the command whose output is to be included within ‘backquotes‘. For example:

```
#!/bin/csh

echo Date and time is:
date
echo
echo "Your username is:" ‘whoami‘
echo "Your current directory is:" ‘pwd‘
```

## Editing several files at once (1)

Suppose we want to change “cie” to “cei” in all files in the current directory whose name ends with “.tex”.

```
#!/bin/csh

ls *.tex | sed ’s/.*/sed s\/cie\/cei\/g & > &tmp/’ > s1
ls *.tex | sed ’s/.*/mv &tmp &/’ > s2
csh s1
csh s2
rm s1 s2
```

This script creates two auxiliary scripts: s1 and s2.

The command “csh” forks a non-interactive C shell process that interprets the commands in a script.

## Editing several files at once (2)

Suppose there are three files in the current working directory whose names end with “.tex”: “1.tex”, “2.tex” and “3.tex”.

Script s1

```
sed s/cie/cei/g 1.tex > 1.textmp
sed s/cie/cei/g 2.tex > 2.textmp
sed s/cie/cei/g 3.tex > 3.textmp
```

Script s2

```
mv 1.textmp 1.tex
mv 2.textmp 2.tex
mv 3.textmp 3.tex
```

## sed scripts

```
grep href publications.html \
  | sed ’s/[ˆ"]*"//’ \
  | sed ’s/".*//’
```

Instead of giving a single editing command on the command line, we can create a script file containing a sequence of editing commands.

pubs.sed

```
s/[ˆ"]*"//
s/".*//
```

We can now request that sed applies the the commands in this script file:

```
unix> grep href publications.html | sed -f pubs.sed
```

## Plotting PDB size with GNUPLOT

```
unix> gnuplot

        G N U P L O T
        Unix version 3.7
        patchlevel 1
        last modified Fri Oct 22 18:00:00 BST 1999

        Copyright(C) 1986 - 1993, 1998, 1999
        Thomas Williams, Colin Kelley and many others

        Type ‘help‘ to access the on-line reference manual
        The gnuplot FAQ is available from
        <http://www.ucc.ie/gnuplot/gnuplot-faq.html>

        Send comments and requests for help to <info-gnuplot@dartmouth.edu>
        Send bugs, suggestions and mods to <bug-gnuplot@dartmouth.edu>


Terminal type set to ’x11’
gnuplot> plot "Deposited" with lines
gnuplot> plot "Released" with lines
gnuplot> plot "Deposited" with lines, "Released" with lines
gnuplot> exit
unix>
```

## get_pdb_stats

```
#!/bin/csh

# Select lines of StatReport.html containing tabular data,
#  | replace all HTML tags by a space character, store result in ’temp’.
#
# Store Year-Deposited pairs (columns 1 and 2) in ’Deposited’.
# Store Year-Released pairs (columns 1 and 3) in ’Released’.
# Remove the temporary file.

grep ’align=right’ /users/mdstud/kemp/ptools/html/StatReport.html \
  | sed ’s/<[ˆ>]*>/ /g’ > temp

awk ’{print $1, $2}’ temp > Deposited
awk ’{print $1, $3}’ temp > Released
rm temp
```

## Controlling interactive programs from within scripts

The interactive email program “mailx” takes input from standard input. To send the contents of “file” to “kemp”, we can use the following command on the command line or within a script:

```
mailx kemp < file
```

Alternatively, we can include the text of a message within a script:

```
#!/bin/csh
mailx kemp << END
This is a test message.
END
```

or:

```
#!/bin/csh
mailx $1 << END
Dear $1,
This is a test message.
END
```