

```
# Linux Tutorial

# prepared for BIFX-550 --Dr. S. Ravichandran, 09/06/2018
# comments lines are shown in green font and start with # (pound) sign

# let us find out where are we in the system?

# Make sure you are in the tcsh by typing

tcsh

# Let us check where we are now (what directory?)

pwd

# What is my username?

whoami

# find out who else have logged in at the moment

who

# let us find out what shell our administrator has assigned us

echo $SHELL

# let us go into tcsh (C-SHELL) using tcsh

tcsh

# Let us find out what shells are available for the users

cat /etc/shells

# What is today's date/time?

date

# calendar

cal 10 1775

# What processes are running?

ps

# Help on one of the command we learned, ls
# Let us also learn how to read man pages. Type "q" (without quotes)
# to quit man pages

man ls

man pwd

# clear the screen

clear
```

```
# based on the help pages, let us add some flags to ls command. Let us
# start with a ls command

ls /usr/local

# use "man ls" (type this without quotes) to identify what flags we
# can use with ls

ls -fla /usr/local

ls -fl

cd /tmp
# find out where you are

pwd
# Let us go home; We can also do this with
# the following alternative, cd ~ or cd $HOME or just cd

cd ~

cd $HOME

# let us create a directory called foo

mkdir foo

# Let us check the permission of the directory using "ls -l" command
ls -l

# let us change the permission of the directory using chmod command
chmod 777 foo

# let us check again to see that the permission of the directory
#has changed

ls -ltr

# And let us go back to our newly created directory, foo

cd foo

# let us check where we are now
pwd

# we can echo a string
echo hello!

# we can save this message in a file called bar
echo hello! > bar
```

```
# check with pwd and ls
pwd

ls -l

# Make sure you are in the foo directory
# let us create a subdirectory called foo2

mkdir foo2

#to go into the foo2, use cd
cd foo2

# check where you are using pwd

pwd

# to go one level up use

cd ../

# make sure you are in foo directory
pwd

# to see the contents of a file, use cat
cat bar

# man is the command for manual pages
# Check to see what the command less does?

#####

# we can remove directories (folders) and files ;
# I am not going to teach you that
# there is no way to get the removed file back.
# let us again make sure you are in foo directory

wget http://files.rcsb.org/download/4hhb.pdb
# let us view the file using nano file editor
nano 4hhb.pdb

cat 4hhb.pdb

# type "q" (without quotes) to quit; press spacebar to move forward
more 4hhb.pdb

head 4hhb.pdb
```

```

# let us copy the file into 4hhbcopy.pdb
cp 4hhb.pdb 4hhbcopy.pdb

# use ls to see that you have two files
ls

# let us remove the file, 4hhbcopy.pdb.
rm -i 4hhbcopy.pdb

# let us clear the screen with "clear"
clear

# let us get one more pdb file
wget http://files.rcsb.org/download/1omp.pdb

# cat command
cat 1omp.pdb

# let us count the lines/words in the 4hhb.pdb
wc -l 4hhb.pdb
wc -w 4hhb.pdb

# let us explore the pipe command
cat 1omp.pdb | grep -i seqres

# let us merge two files
cat 4hhb.pdb 1omp.pdb > 4hhb-1omp.pdb

grep -i Fe 4hhb.pdb

# what commands we have typed so far
history

# which program; finding an executable file with which
# has to be in your path
which ls

# Find out the difference between which and whereis
whereis pwd

# Go the following link,
# https://www.ncbi.nlm.nih.gov/protein/NP\_000509.1?report=fasta, and
# cut-and-paste the following sequence into a file called
# np_000509.fasta. For this step you can use the nano editor.

```

- a) first open a nano editor with the following command
- b) nano np_000509.fasta
- c) paste the sequence file
- d) Use control-x to save and quit

```
>NP_000509.1
MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDVAVMGNPKVKAHGKKVLG
AFSDGLAHLNLDNLKGTFTLSELHCDKLHVDPENFRLLGNVLCVLAHHFGKEFTPPVQAAYQKVVAGVAN
ALAHKYH
```

```
# Let us explore the sequence file using Linux commands
```

```
od -b np_000509.fasta # first col is the byte offset
```

```
od -c np_000509.fasta
```

```
# let us create a file
```

```
nano list.txt
```

```
# and type the following each in one line
```

```
rat chicken monkey chimp cow ecoli human dog
```

```
# save the file and exit
```

```
# when you come out of nano editor.
```

```
# Let us check whether the file, list.txt exists
```

Your TURN

```
# let us view the file,
```

Your TURN

```
# let us sort the file and save it in sortedlist.txt
```

```
sort < list.txt > sortedlist.txt
```

```
ls *list
```

```
#Explore the following commands on your own
```

```
chmod
```

```
ps
```

```
sleep
```

```
df
```

```
du
```

```
quota
```

```
gzip
```

```
zcat
```

```
zcat
```

file

diff

find

logging into other computers

ssh username@pronto.abc.com

creating links

ln -s perl5.005_03 perl

#filename conventions fortran: f90, C: c; C++: cc

test.f90

test.c

test.cc

test.CC

SHELL Tutorial

In this tutorial, I am going to explain some basic TCSH commands.

What is a SHELL?

How it can be useful?

Why I should learn at least one SHELL ?

#In the following tutorial, we are going to download 4 PDB structure
#files from www.rcsb.org . This process can be done for any number of
#files, for now, we are going to explain this using just 4 files.

make sure we are in tcsh by typing before this exercise

tcsh

foreach f (1stp 7lyz 2src 3v3k 4IGK 4JLU)

wget [http://files.rcsb.org/download/\\$f.pdb](http://files.rcsb.org/download/$f.pdb)

echo \$f

end

```
#Can we use shell programming to count the number of lines of the  
# files (Think if you can do this for millions of files?!)
```

```
foreach f (*.pdb)  
    sed -e 's/REMARK/Remark/g' < $f > M$f  
    echo "Done"
```