Shell Basics cheat sheet

Created by OU ACM

Navigation

ls ***dir***
list all files in *dir* (optional)

cd ***dir***
go to directory *dir*

pwd
print working directory

cat ***file***
output contents of *file*

Symbols

.
current directory

..
parent directory

command **>** ***file***
replace contents of *file* with output of *command*

command **>>** ***file***
append output of *command* to *file*

command **<** ***file***
use contents of *file* as input to *command*

commandA **|** ***commandB***
use output of *commandA* as input to *commandB*

*
wildcard to match any string

?
wildcard to match a single character

File Analysis

diff ***file1 file2***
output differences between *file1* and *file2*

wc ***file***
count the lines, words, and bytes in *file*

grep ***searchFor files***
find all instances of string (or regex) *searchFor* in *files*

File Manipulation

mkdir ***dir***
make directory *dir*

touch ***file***
create *file*

mv ***file newfile***
rename/move *file* to *newfile*

cp ***file newfile***
copy *file* to create *newfile*

rm ***file***
delete *file*

Misc Useful

echo ***strings***
output *strings*

man ***command***
open manual to *command*

clear
clear the terminal

which ***command***
find location of *command*

chmod u+x ***file***
allow *file* to be executed by owner

Remote Connection

ssh ***user@host***
open secure remote shell to *host*

scp ***file user@host:remotefile***
copy *file* to remote *remotefile* on *host*

scp ***user@host:remotefile file***
copy *remotefile* from *host* to local *file*

Stocker Lab Computers:
computer**.cs.ohio.edu**
pu1, pu2, pu3
odd05, odd07, ... odd39, odd41

Expansions	*Examples*
prestring{values}poststring creates n strings, replacing *{values}* with each comma-separated *value*	***echo hello-{a,bc}-world*** *> hello-a-world hello-bc-world*
VAR=value set variable *VAR* to *value*	***MYVAR=”hello”*** ***echo $MYVAR*** *> hello*
$VAR ${VAR} replace with value of variable *VAR*	***echo ${MYVAR}*** *> hello*
$(*command*) `*command*` replace with output of *command*	***echo $(pwd) `ls`*** *> Users/user/dir file1 file2 file3*
$((*arithmetic*)) replace with solved math of *arithmetic*	***echo $((5+3*4))*** *> 17*

If Statements

if [[*conditionalA*]]; then *commandA*;
elif [[*conditionalB*]]; then *commandB*;
else *commandC*; fi
if / else if / else statement

[[*op string*]]
check if length of string is zero (**-z**)
or not (**-n**)

[[*string1 op string2*]]
check if *string1* and *string2* are equal
(**=**) or not equal (**!=**)

[[*num1 op num2*]]
compare *num1* to *num2* using an *op* value:

- **-eq** equal
- **-ne** not equal
- **-lt** less than
- **-le** less than or equal to
- **-gt** greater than
- **-ge** greater than or equal to

C++ Compiler

g++ *options file*
compile C++ code *file* with *options*:

-Wall give all warnings
-Werror make all warnings errors
-o *exeName* compile to executable *exeName*

Note: this covers only the basics of Shell. There are many more commands, and the ones listed have much more functionality. Explore documentation and cli man pages to learn more

Loops

while [[*conditional*]];
do *commands*;
done
while loop

for *var* in *strings*;
do *commands*;
done
for loop to iterate through *strings*

for ((*var* = *start*; *var* < *end*; *var* ++));
do *commands*;
done
for loop to iterate across numbers

Git

git clone *url*
clone repo from GitHub to local machine

git pull
update local repo with changes on GitHub

git add *file*
stage *file* to be committed (use **-A** to
stage all updated files)

git commit -m *message*
commit all staged files

git push
push committed changes to GitHub

git checkout *branch*
go to *branch* (use **-b** to make new branch)