

BS859 SCC guide

Directories and Paths

<i>Path</i>	<i>Description</i>
rcs.bu.edu scc-ondemand.bu.edu	Research Computing Services website SCC OnDemand
<code>~</code>	Home Directory (Note: this is an alias to a real path which looks like <code>/usr4/groupname/username</code>).
<code>/projectnb/bs859/students/username</code>	Your working directory for BS859 course (substitute <i>username</i> with your BU username.)
<code>/projectnb/bs859/students/materials/</code>	The materials directory for BS859 course. This directory will contain code for the computer lab portion of each class.
<code>/projectnb/bs859/students/data/</code>	The BS859 Data directory.

Linux Commands

<i>command</i>	<i>description</i>
<code>pwd</code>	Display current directory
<code>ls</code>	list all files and sub-directories
<code>ls -l</code>	display all information about files and sub-directories in the current directory
<code>ls -l /projectnb/bs859/students/data</code>	list files in a specific directory
<code>cd /projectnb/bs859/students/username</code>	Change current directory to your project bs859 directory (specifying the absolute path)
<code>mkdir lab1</code>	create a new sub-directory <i>lab1</i>
<code>cd lab1</code>	Change current directory to a sub-directory <i>lab1</i> (specifying a relative path)
<code>cd ..</code>	Go to a parent directory
<code>cd</code>	Go to the home directory
<code>cp /projectnb/bs859/students/materials/file1.txt .</code>	copy <i>file1.txt</i> to the current directory (dot is important!)
<code>cp /projectnb/bs859/students/materials/*.txt .</code>	copy all the files with a <i>txt</i> extension to the current directory
<code>mv file1 file2</code>	rename (move) file <i>file1</i> to file <i>file2</i>
<code>rm filename</code>	delete file
<code>rm -rf dirname</code>	delete <i>dirname</i> with all sub-directories and files in it
<code>gedit myfile.sh</code>	Open <i>myfile.sh</i> script for editing.
<code>export MYVAR=/some/string/or/value</code>	Create an environment variable
<code>echo \$MYVAR</code>	Print the value of an environment variable
<code>head myfile</code>	display first 10 lines of <i>myfile</i>
<code>head -n 20 myfile</code>	display first 20 lines of <i>myfile</i>
<code>tail myfile</code>	display last 10 lines of <i>myfile</i>
<code>wc myfile</code>	Print the number of lines, words, and symbols in <i>myfile</i>
<code>wc -l myfile</code>	Print only the number of lines in <i>myfile</i>
<code>ls -l > myfile</code>	Redirect the output of a command to be saved in <i>myfile</i>
<code>grep chr22 myfile</code>	Search the word <i>chr22</i> in <i>myfile</i>
<code>ls -l grep myfile</code>	An example of a linux <i>pipe</i> (similar to <code>%>%</code> in R): In the output of <code>ls -l</code> command search for <i>myfile</i>

Script file syntax

```
#!/bin/bash -l

# Display my current directory
pwd

# Set environment variable
export MYDIR=/projectnb/bs859/students/ktrn

# List files in the directory specified in MYDIR
ls -l $MYDIR
```

Traditionally, bash scripts have extension *.sh*, e.g.: *script.sh*. To execute this script:

```
bash script.sh
```

Batch Job Script file syntax

```
#!/bin/bash -l

# Specify project
#$ -P bs859

# Join error and output streams
#$ -j y

# Send yourself an email when job is done
#$ -m y

module load R/4.1.1
Rscript myscript.R
```

Job submission scripts may have extension *.qsub* or *.sh*: *script.qsub*. To submit a job:

```
qsub script.qsub
```

Check the status of your job:

```
qstat -u username
```