Using GitHub Classroom

BBM103 Introduction to Programming Lab 1

Fall 2017
Signing Up to GitHub Classroom
Signing Up to GitHub Classroom

Click to sign up unless you have an educational account.

Fill textboxes and click sign in button to authorize.
Signing Up to GitHub Classroom

You MUST create your account with your IDs beginning with ‘b’.
Signing Up to GitHub Classroom

There are two options. We recommend that you choose the 1st option unless you need a private repository.
Signing Up to GitHub Classroom

Open your mailbox to verify your github account.

Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.
An email containing verification instructions was sent to your email address.

Didn’t get the email? Resend verification email or change your email settings.
Signing Up to GitHub Classroom

Click the link provided within the mail content
Signing Up to GitHub Classroom

Learn Git and GitHub without any code!

Using the Hello World guide, you’ll create a repository, start a branch, write comments, and open a pull request.

Read the guide  Start a project
Signing Up to GitHub Classroom
Joining BBM103 Classroom

Now authorize github account.
Joining BBM103 Classroom

You should accept the assignment activated to push(submit) your works.

Assignment title
Joining BBM103 Classroom

Accepted the test2 assignment

You are ready to go!

You may receive an invitation to join via email invitation on your behalf. No further action is necessary.

Your assignment has been created here: https://github.com
Joining BBM103 Classroom

Quick setup — if you’ve done this kind of thing before

- Set up in Desktop
- HTTPS: https://github.com/

We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

- echo "# README.md" > README.md
- git init
- git add README.md
- git commit -m "first commit"
- git remote add origin https://github.com/
- git push -u origin master
How to Use the Linux Command Line

BBM103 Introduction to Programming Lab I

Fall 2017
The Shell & Terminal

- **The Shell** is a program that takes commands from the keyboard and gives them to the operating system to perform.

- **Terminal Emulator** is a program that opens a window and lets you interact with the shell.
Basic Commands

• When you open a terminal emulator, by default you are in the home directory of the logged in user.

• You will see the name of the logged in user followed by the hostname.
  
  • $ means you are logged in as a regular user
  
  • # means you are logged in as root.
pwd

- **pwd** prints the full path of your current working directory.

```
[selimy@rdev ~]$ pwd
/home/akd/selimy
[selimy@rdev ~]$
```
You can list all directories and files inside the current directory by using the `ls` (or `ls -l; ll` for listings including information such as the owner, size, date last modified and permissions) command.
**cd**

- The `cd` command is used to change the current directory.

```
[selma@rdev test]$ ls
some_directory_1  some_directory_2
[selma@rdev test]$ cd some_directory_1
[selma@rdev some_directory_1]$ ...
```

- To change to the parent of the current directory use `cd ..`

```
[selma@rdev some_directory_1]$ cd ..
[selma@rdev test]$ ...
```

- To return directly to the home directory use a tilde as the argument:

```
[selma@rdev test]$ cd ~
[selma@rdev ~]$ ...
```
Manipulating Files

• **cp** - copy files and directories
• **rm** - remove files and directories
• **mv** - move or rename files and directories
• **mkdir** - create directories
• **cat** - create new file, concatenate files
• `cp` copies files and directories. In its simplest form, it copies a single file:
cp (cont.)

- You can specify the full path to where you want to copy your file:

```
[necva@rdev ~]$ ls
bbm103  but    error    Maildir    output
bm104  cloud  Graph-Cluster monopoly public_html
bm3    cloud.old HelloWorld.py Morpheme-Graphcluster Word2VecJava-master
[necva@rdev ~]$ cp HelloWorld.py Hello.py
[necva@rdev ~]$ ls
bbm103  cloud    Hello.py    Morpheme-Graphcluster
bm104  cloud.old HelloWorld.py output
bm3    error    Maildir    public_html
but    Graph-Cluster monopoly    Word2VecJava-master
[necva@rdev ~]$ cp HelloWorld.py bbm103/HelloWorld.py
[necva@rdev ~]$ cd bbm103
[necva@rdev bbm103]$ ls
HelloWorld.py
[necva@rdev bbm103]$  
```
If you want to delete any file or directory the command is 'rm' (for files) and 'rm -r' (for directories).

```
[necva@rdev ~]$ ls
bbm103  but    error   Maildir          output
bm104  cloud  Graph-Cluster monopoly    public_html
bm3    cloud.old  HelloWorld.py Morpheme-Graphcluster Word2VecJava-master
[necva@rdev ~]$ rm -r bbm103
[necva@rdev ~]$ ls
bm104  cloud  Graph-Cluster monopoly    public_html
bm3    cloud.old  HelloWorld.py Morpheme-Graphcluster Word2VecJava-master
but    error   Maildir          output
[necva@rdev ~]$
```
mv

- **mv** command moves or renames files and directories depending on how it is used.

```bash
[necva@rdev ~]$ mv Hello.py bbm103
[necva@rdev ~]$ ls
bbm103  but  error  monopoly  public_html
bm104  cloud Graph-Cluster Morpheme-Graphcluster Word2VecJava-master
bm3  cloud.old Maildir output
[necva@rdev ~]$ cd bbm103
[necva@rdev bbm103]$ ls
Hello.py
[necva@rdev bbm103]$ ls
```
mkdir

• If you want to create new directories the command is `mkdir`. 

```
[necva@rdev ~]$ mkdir bbm103
[necva@rdev ~]$ ls
bbm103  but  error  Maildir  output
bm104  cloud  Graph-Cluster  monopoly  public_html
bm3  cloud.old  HelloWorld.py  Morpheme-Graphcluster  Word2VecJava-master
```

10/4/2017
cat stands for Concatenate (birleştirmek). It is used to create new file (with or without content), concatenate files and display the output of files on the standard output.
zip & unzip

- **zip** and **unzip** commands create and extract zip archive files respectively.

```bash
[selim@rdev BBM103Linux]$ cd testDir/
[selim@rdev testDir]$ ls
bashsc.sh  bashsc.sh.bak
[selim@rdev testDir]$ unzip test.zip -d testDir
Archive:  test.zip
   inflating:  testDir/bashsc.sh
   inflating:  testDir/bashsc.sh.bak
[selim@rdev BBM103Linux]$ ls
bashsc.sh  bashsc.sh.bak  testDir  test.zip
[selim@rdev BBM103Linux]$ cd testDir/
[selim@rdev testDir]$ ls
bashsc.sh  bashsc.sh.bak
[selim@rdev testDir]$
```
• The * character serves as a "wild card" for filename expansion. By itself, it matches every filename in a given directory.
Most executable programs intended for command line use provide a formal piece of documentation called a *manual* or *man page*. A special paging program called *man* is used to view them.

```
[necva@rdev bbm103]$ clear
[necva@rdev bbm103]$ man ls
LS(1) User Commands LS(1)

NAME
ls - list directory contents

SYNOPSIS
ls [OPTION]... [FILE]...

DESCRIPTION
List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.

-a, --all
  do not ignore entries starting with .

-A, --almost-all
  do not list implied . and ..
```
ssh

• **ssh** (Secure Shell client) is a program for logging into a remote machine and for executing commands on a remote machine.

```
$ ssh cemil@dev.cs.hacettepe.edu.tr
cemil@dev.cs.hacettepe.edu.tr's password:
```
scp

- scp allows files to be copied to, from, or between different hosts. It uses ssh for data transfer and provides the same authentication and same level of security as ssh.

A simple example that illustrates how to send a file to dev space.

```bash
scp <localfile> <username>@dev.cs.hacettepe.edu.tr:/home/ogr/b****/<directory>
```

celim@selim-PC:~$ scp DPSO.pdf selimy@dev.cs.hacettepe.edu.tr:/home/akd/selimy/
selimy@dev.cs.hacettepe.edu.tr's password:
About chmod

• **chmod** is used to change the permissions of files or directories.
• Example: chmod 777 myFile

<table>
<thead>
<tr>
<th>#</th>
<th>Permission</th>
<th>rwx</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>read, write and execute</td>
<td>rwx</td>
</tr>
<tr>
<td>6</td>
<td>read and write</td>
<td>rw-</td>
</tr>
<tr>
<td>5</td>
<td>read and execute</td>
<td>r-x</td>
</tr>
<tr>
<td>4</td>
<td>read only</td>
<td>r--</td>
</tr>
<tr>
<td>3</td>
<td>write and execute</td>
<td>-wx</td>
</tr>
<tr>
<td>2</td>
<td>write only</td>
<td>-w-</td>
</tr>
<tr>
<td>1</td>
<td>execute only</td>
<td>--x</td>
</tr>
<tr>
<td>0</td>
<td>none</td>
<td>---</td>
</tr>
</tbody>
</table>
Exercise

- All tasks must be performed using linux commands:
  1) Make a new directory named `playing_with_linux_cmd`
  2) Change your current working directory to the newly created one.
  3) List the contents of this directory to see that it is empty.
  4) Create a new text file `jibberish.txt` and write something funny in it before closing it.
  5) Create another new text file `README.txt` and write your life motto in it.
  6) Copy `jibberish.txt` into a text file named `wise_sayings.txt`
  7) Delete `jibberish.txt`
  8) Print out the content of `wise_sayings.txt`
  9) Create a new directory named `my_precious` and move `wise_sayings.txt` into that newly created directory. List the content of the current working directory to make sure that you have successfully moved the file.
  10) Change the permission of the file `wise_sayings.txt` to read, write and execute.
  11) Change your working directory to the parent directory of `playing_with_linux_cmd`
  12) Zip `playing_with_linux_cmd` as `gameover.zip`
  13) Use `scp` command to copy this zipped folder from your local computer to your home directory on our remote server `dev.cs.hacettepe.edu.tr`