

Linux System Guide for CS20300 Students

Original Version by Taekyung Heo

Updated by Soojin Hwang

Updated by Jinwon Pyo

Updated by Hongbeen Kim

What is Linux?

- **What is Linux?**

- Linux is a modern, free operating system.
- First developed by Linus Torvalds in 1991.
- **Features**
 - Multi-tasking, multi-user
 - Various distributions (Ubuntu, CentOS, ...)
 - Fully customizable

- **Why do we need Linux?**

- Knowing how to use Linux systems is important for computer engineers.
- You will do your projects on Linux systems.

Connection to a Linux Machine

- How can I use a Linux machine?
 - You have to establish a connection to it.
- How can I connect to a machine?
 - Use ssh clients like PuTTY or **VSCode**.



PuTTY

PuTTY: A Free Telnet/SSH Client

- Very light-weight ssh client
- Does not need installation
- Not user-friendly interface



VSCode

VSCode (Visual Studio Code)

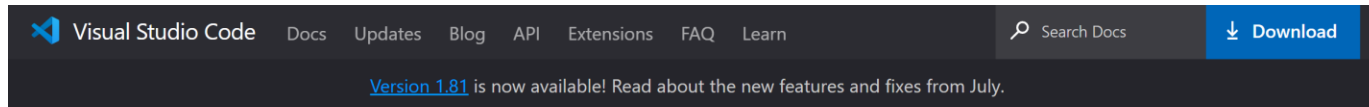
- Needs installation & setup
- User-friendly interface

**We will
exclusively
use VSCode
in this lab!**

Install VSCode

- First You need to install VSCode

- Link: <https://code.visualstudio.com/download>
- Choose the OS version accordingly
- Install the VSCode (you can just say 'ok' or 'yes' for installation options during the installation process)

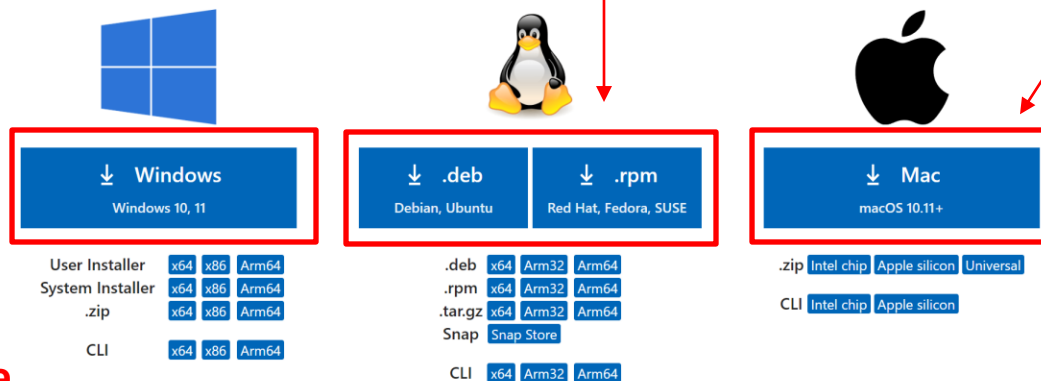


Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

**Choose it if you are
using Linux family OSes**

**Choose it if you are
using MacOS**



**Choose it if you are
using Windows**


Connection Example: VSCode

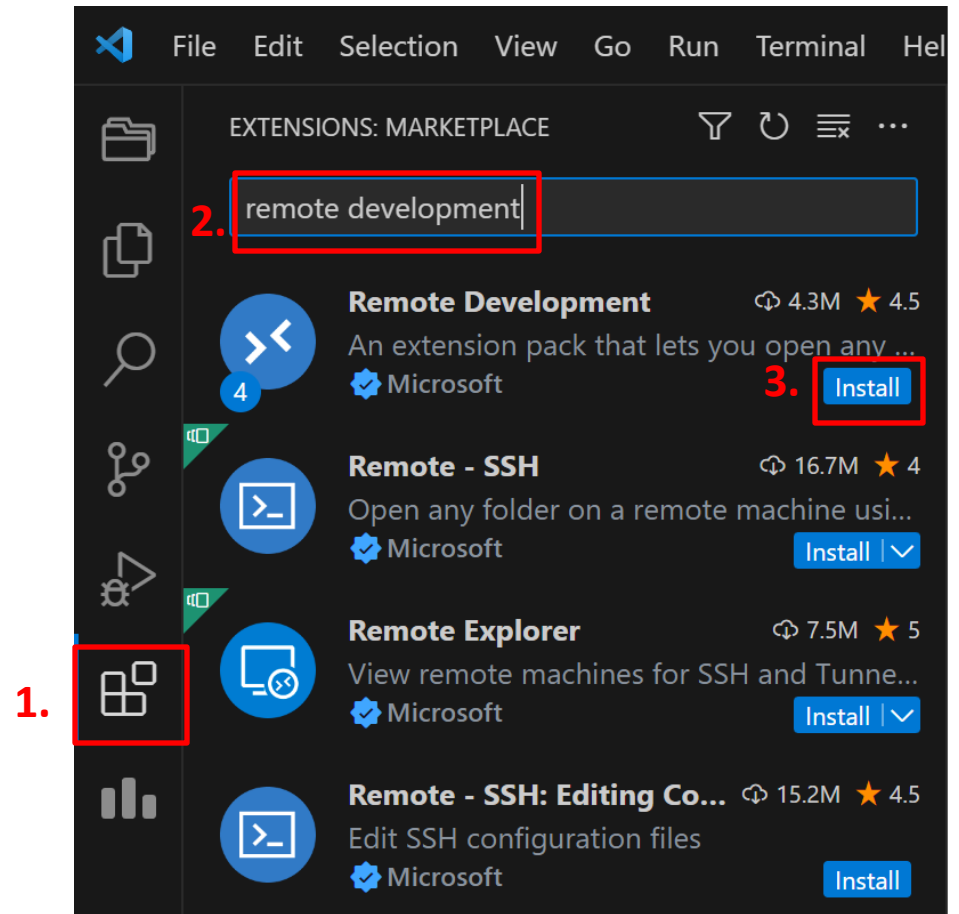
Assuming you have already installed VSCode in your local machine

Let's make a connection to the server machine to which you are designated

First you need to install one VSCode extension

Steps

1. Click the icon 
2. Search "remote development"
3. Install it



Connection Example: VSCode - 2

Now we are ready to make a SSH-connection to your designated remote machine

e.g.) Connection information

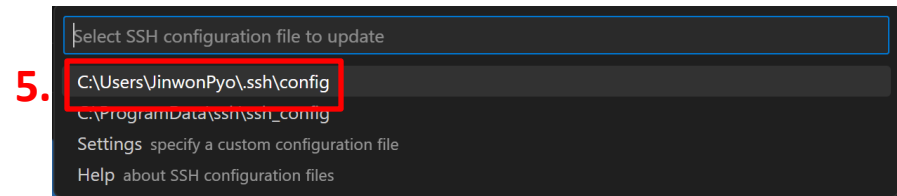
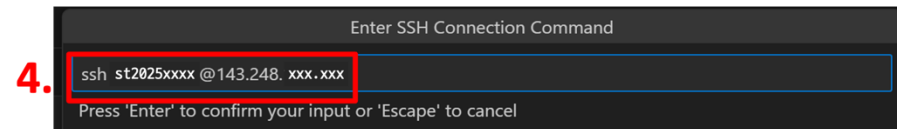
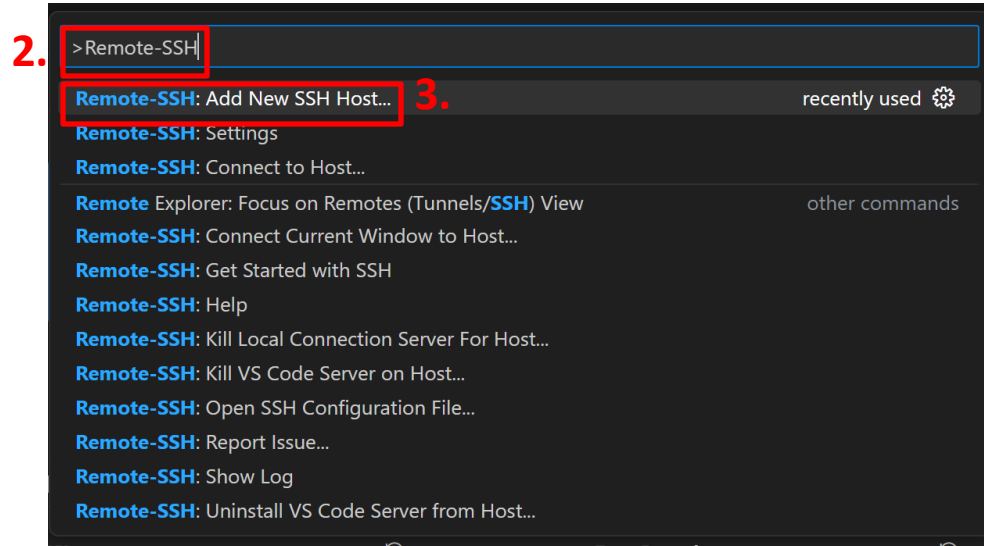
IP address: Your assigned IP address

ID: st[your student id]

Password: Your assigned password

Steps.

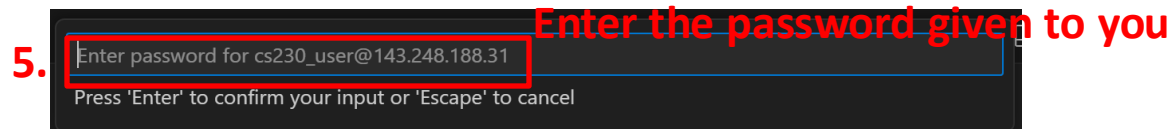
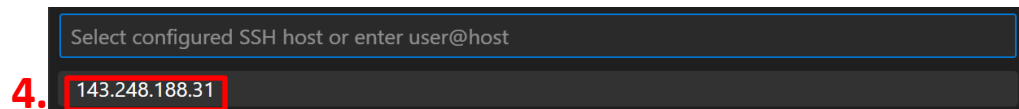
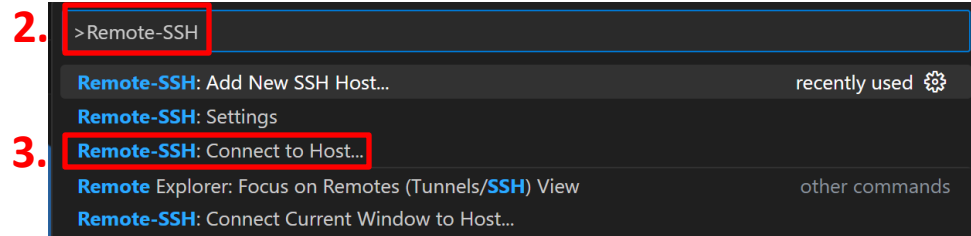
1. Hit the “F1” key (dropdown menu will pop)
2. Search “Remote-SSH”
3. Select “Add New SSH Host...”
4. Enter the given information as the following format (**When asked to choose the targeting OS, then choose “Linux”**)
5. Just select the first option



Connection Example: VSCode - 3

Finally let's make an actual SSH connection to your remote machine

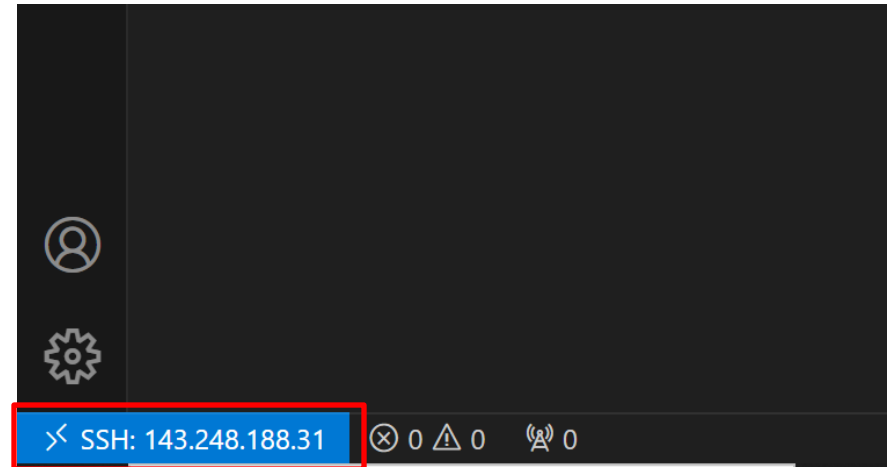
1. Hit the “F1” key (dropdown menu will pop)
2. Search “Remote-SSH”
3. Select “Connect to Host...”
4. Select the IP address you entered during the previous step
5. New VSCode window will pop out and ask you to enter password



Connection Example: VSCode - 4

You are now in the remote machine!

Check out the bottom left corner of your VSCode window

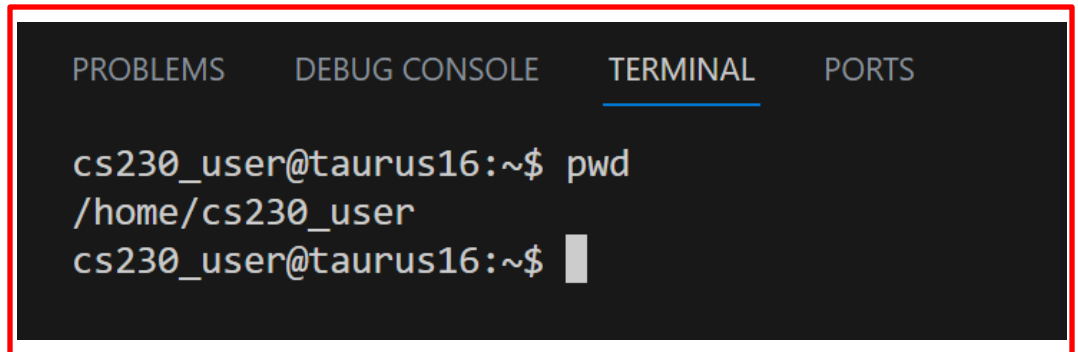
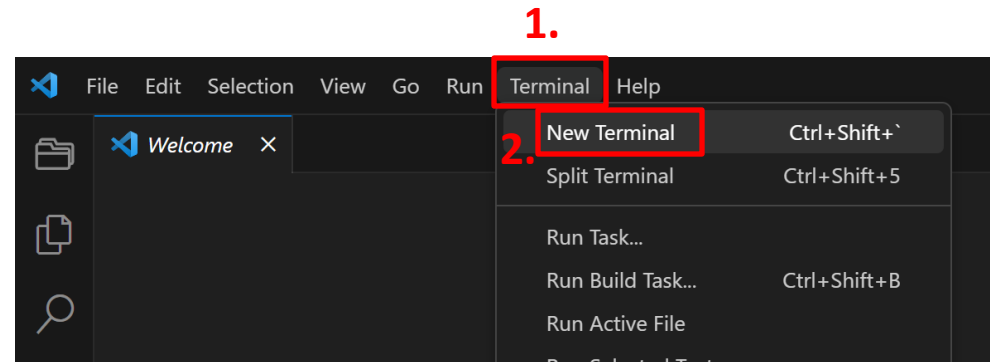


Connection Example: VSCode - 5

You can also open up the Linux shell (or terminal) in VSCode through which you can interact your remote machine

Steps.

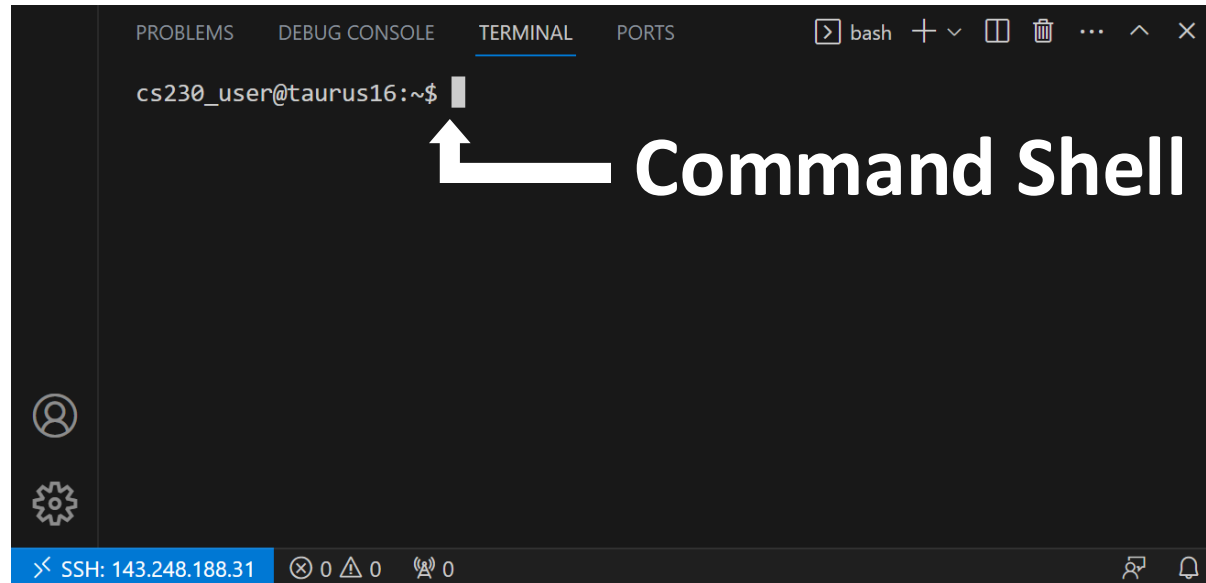
1. Find “Terminal”
2. Select “New Terminal”
3. Then you will see the Linux shell pop up in the bottom of VSCode window



3. This is the Linux shell

Linux Shell

Connected!



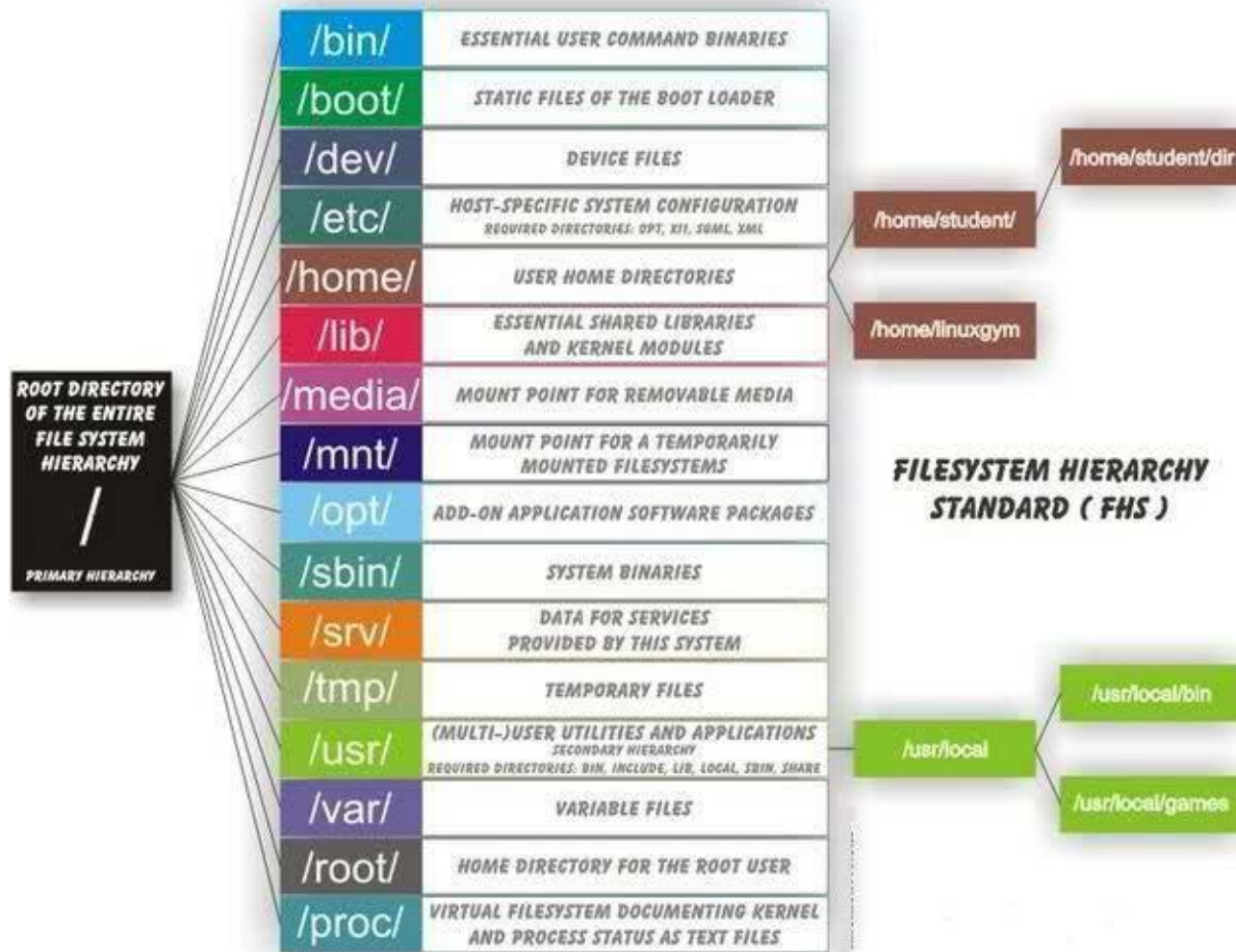
Shell?

- a command line interface between a user and Linux machine

Linux Directory

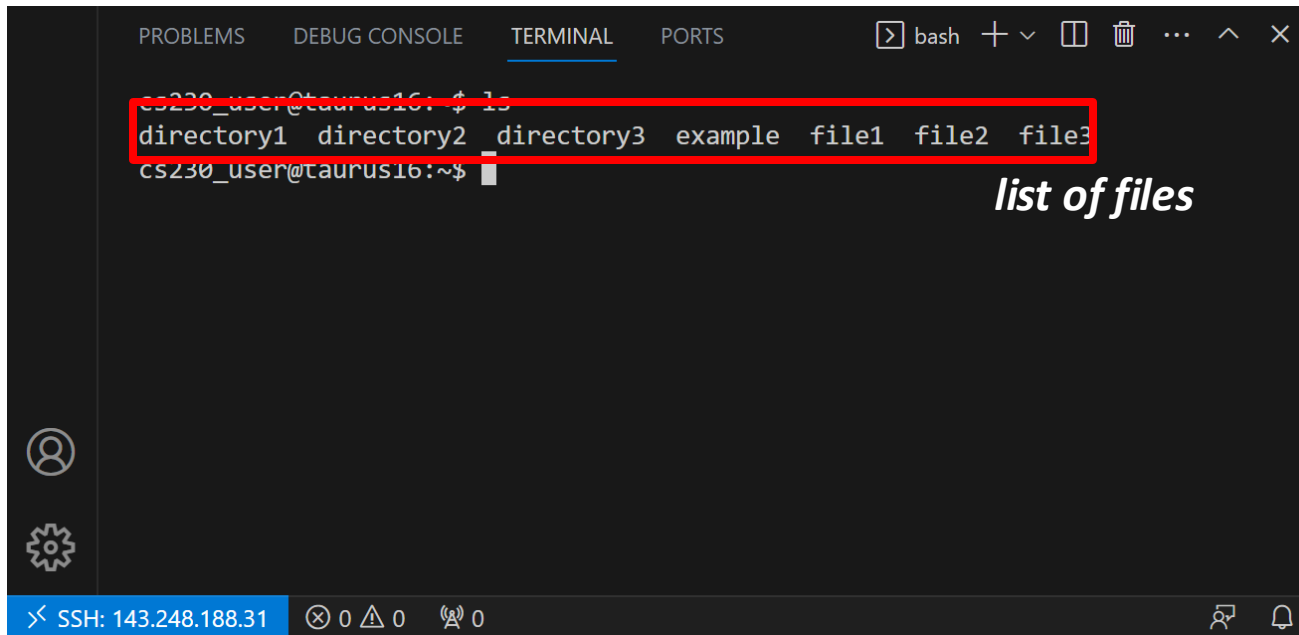
- / : root directory
- ~ : user's home directory (usually same as /home/[username])
- . : current(working) directory
- .. : upper(parent) directory

Linux Directory Structure



Linux Commands

- ls [directory] (empty for working directory)
 - list directory contents



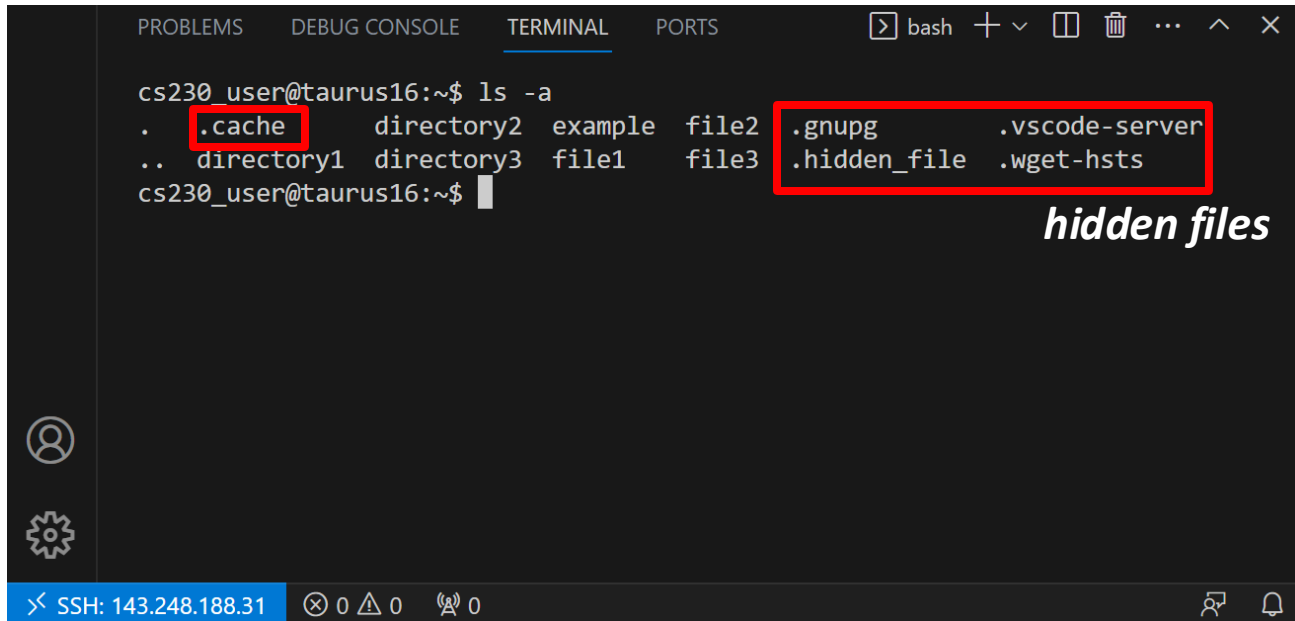
A screenshot of a terminal window with a dark background. The terminal title bar shows tabs for 'PROBLEMS', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. On the right side of the title bar are icons for a terminal window, a plus sign, a dropdown arrow, a trash can, and a close button. The terminal content shows a prompt 'cs230_user@taurus16: ~\$' followed by the command 'ls'. The output of the command is 'directory1 directory2 directory3 example file1 file2 file3', which is highlighted by a red rectangular box. Below the output, the prompt 'cs230_user@taurus16:~\$' is visible. To the right of the terminal output, the text '*list of files*' is written in a white, italicized font. On the left side of the terminal window, there are icons for a user profile and a gear. At the bottom of the window, a blue status bar displays 'SSH: 143.248.188.31' and some system icons on the right.

```
cs230_user@taurus16: ~$ ls
directory1 directory2 directory3 example file1 file2 file3
cs230_user@taurus16:~$
```

list of files

Linux Commands

- ls [directory] (empty for working directory)
 - -a : Print the list including hidden contents (. name)



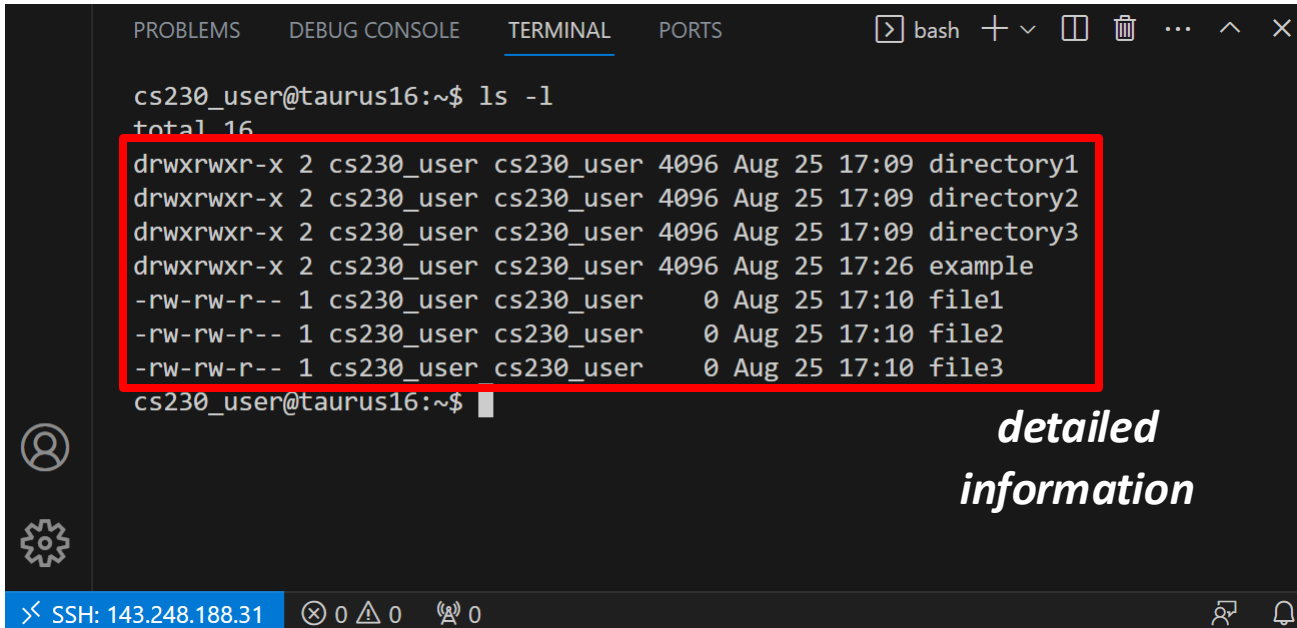
```
cs230_user@taurus16:~$ ls -a
.  .cache  directory2  example  file2  .gnupg  .vscode-server
.. directory1 directory3  file1    file3  .hidden_file .wget-hsts
cs230_user@taurus16:~$
```

hidden files

SSH: 143.248.188.31

Linux Commands

- `ls [directory]` (empty for working directory)
 - `-l` : Print the list of files with detailed information
 - Permissions, number of links, owner name, owner group, size, last modification time, name



```
PROBLEMS  DEBUG CONSOLE  TERMINAL  PORTS
bash + - [ ] [ ] ... ^ x

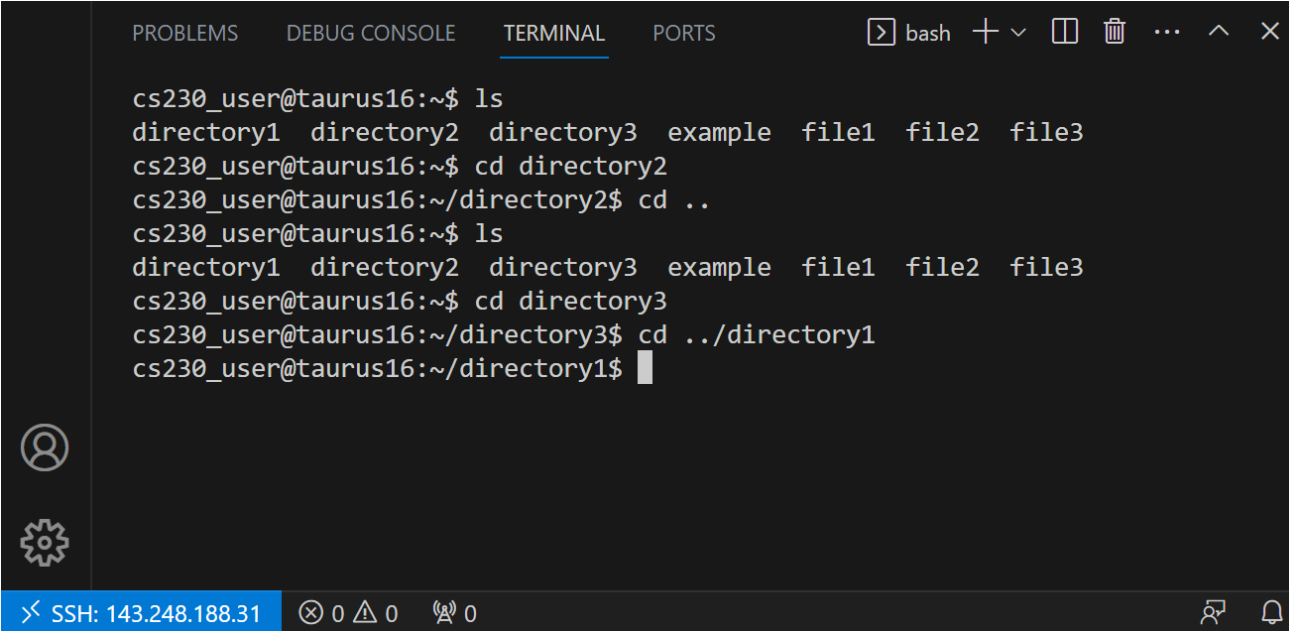
cs230_user@taurus16:~$ ls -l
total 16
drwxrwxr-x 2 cs230_user cs230_user 4096 Aug 25 17:09 directory1
drwxrwxr-x 2 cs230_user cs230_user 4096 Aug 25 17:09 directory2
drwxrwxr-x 2 cs230_user cs230_user 4096 Aug 25 17:09 directory3
drwxrwxr-x 2 cs230_user cs230_user 4096 Aug 25 17:26 example
-rw-rw-r-- 1 cs230_user cs230_user  0 Aug 25 17:10 file1
-rw-rw-r-- 1 cs230_user cs230_user  0 Aug 25 17:10 file2
-rw-rw-r-- 1 cs230_user cs230_user  0 Aug 25 17:10 file3
cs230_user@taurus16:~$
```

*detailed
information*

SSH: 143.248.188.31 0 0 0

Linux Commands

- `cd [directory]`
 - change working directory



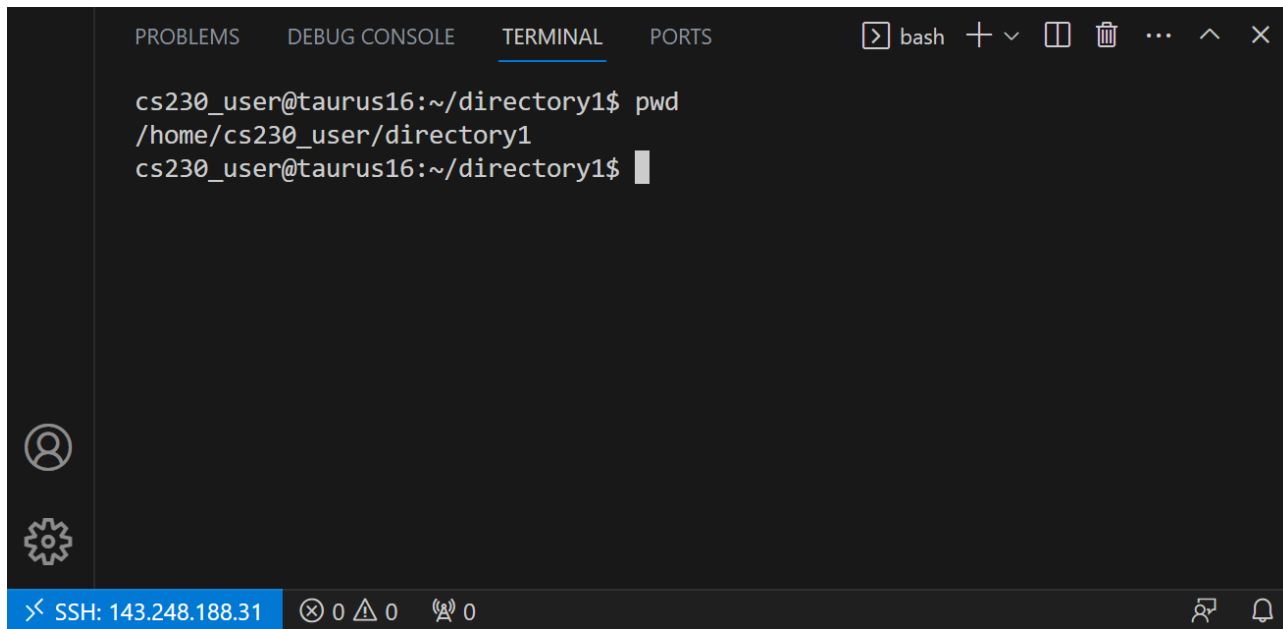
The screenshot shows a terminal window with a dark background. The title bar at the top includes tabs for 'PROBLEMS', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. To the right of the tabs are icons for a terminal window, a plus sign, a dropdown arrow, a trash can, and a close button. The terminal content shows a user named 'cs230_user' on a machine named 'taurus16' performing a series of directory operations. The prompt is '~\$'. The commands and their outputs are as follows:

```
cs230_user@taurus16:~$ ls
directory1 directory2 directory3 example file1 file2 file3
cs230_user@taurus16:~$ cd directory2
cs230_user@taurus16:~/directory2$ cd ..
cs230_user@taurus16:~$ ls
directory1 directory2 directory3 example file1 file2 file3
cs230_user@taurus16:~$ cd directory3
cs230_user@taurus16:~/directory3$ cd ../directory1
cs230_user@taurus16:~/directory1$
```

On the left side of the terminal window, there are two icons: a user profile icon and a gear icon for settings. At the bottom of the window, a status bar shows 'SSH: 143.248.188.31' on the left, and in the center, it displays connection statistics: '0' for errors, '0' for warnings, and '0' for a third metric. On the right side of the status bar, there are icons for a chat window and a notification bell.

Linux Commands

- pwd
 - print name(absolute path) of working directory

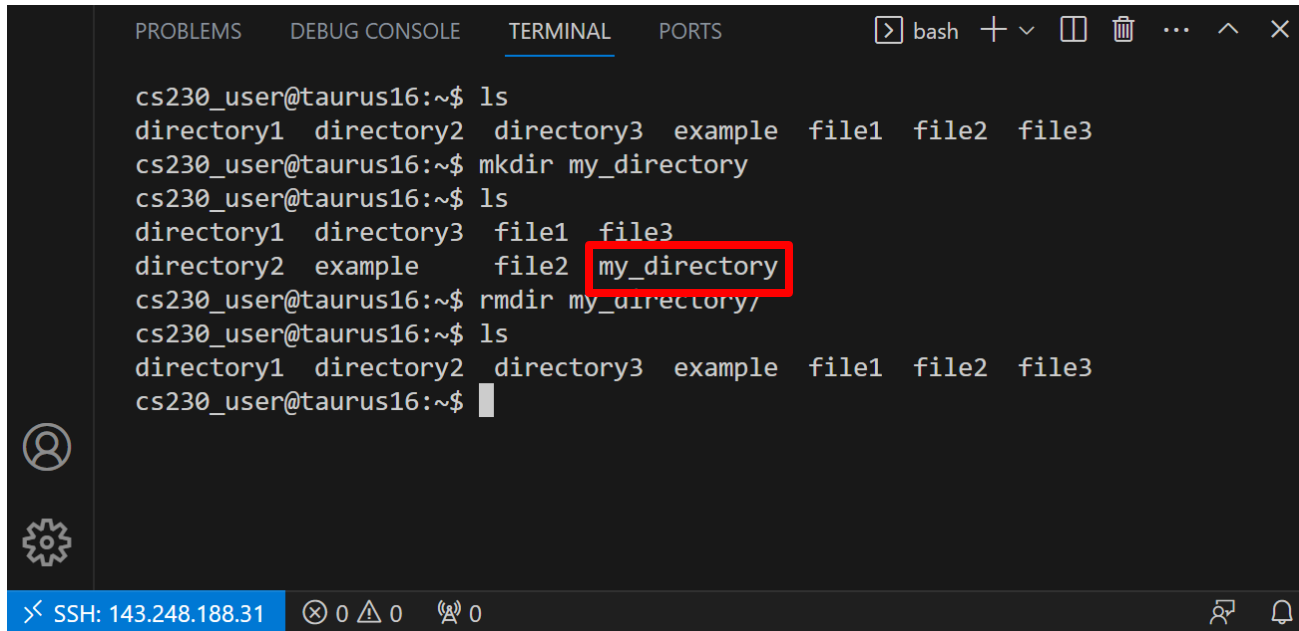


A screenshot of a terminal window with a dark background. The window has tabs at the top labeled 'PROBLEMS', 'DEBUG CONSOLE', 'TERMINAL' (which is selected), and 'PORTS'. To the right of the tabs are icons for a terminal window, a plus sign, a dropdown arrow, a window icon, a trash icon, an ellipsis, an up arrow, and a close button. The terminal content shows the prompt 'cs230_user@taurus16:~/directory1\$' followed by the command 'pwd'. The output of the command is '/home/cs230_user/directory1'. Below the output, the prompt 'cs230_user@taurus16:~/directory1\$' is shown again with a cursor. On the left side of the terminal window, there are two icons: a user icon and a gear icon. At the bottom of the window, there is a status bar with a blue section on the left containing the text '> SSH: 143.248.188.31', followed by three status indicators: a red 'x' with '0', a yellow triangle with '0', and a green 'A' with '0'. On the far right of the status bar are icons for a terminal window and a bell.

```
cs230_user@taurus16:~/directory1$ pwd
/home/cs230_user/directory1
cs230_user@taurus16:~/directory1$
```

Linux Commands

- mkdir [name], rmdir [name]
 - mkdir – make directories
 - rmdir – remove empty directories



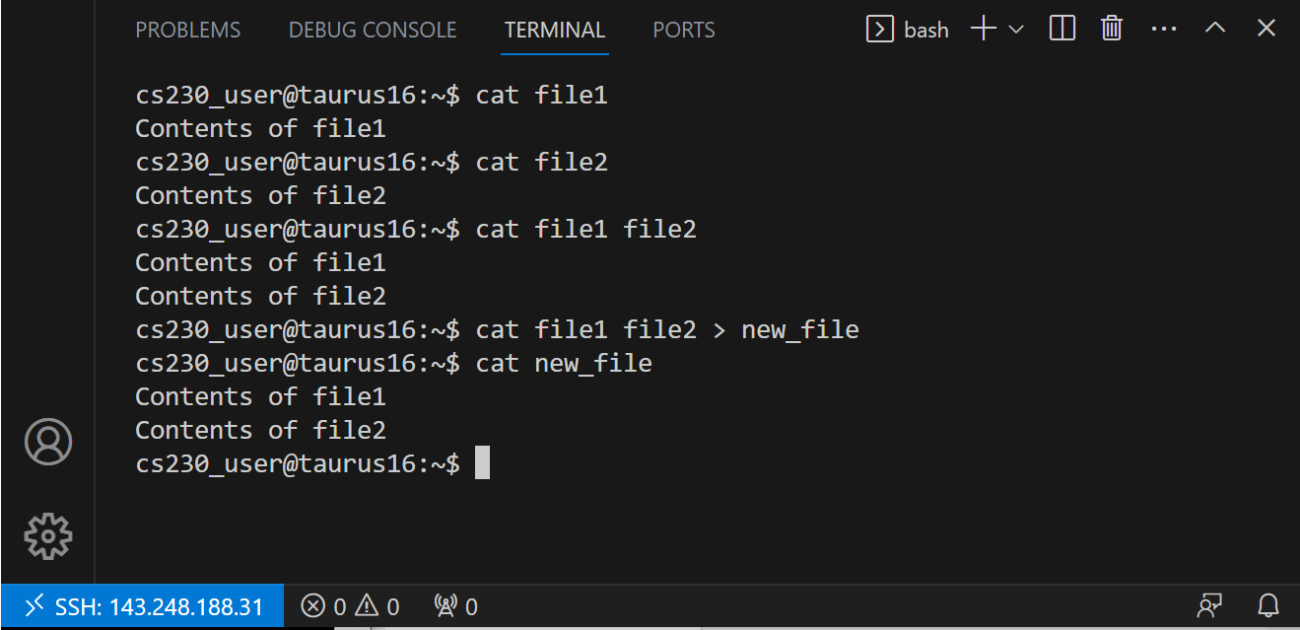
A screenshot of a terminal window with a dark background. The terminal shows a series of commands and their outputs. The prompt is `cs230_user@taurus16:~$`. The first `ls` command lists `directory1`, `directory2`, `directory3`, `example`, `file1`, `file2`, and `file3`. Then, `mkdir my_directory` is executed. A second `ls` command shows the same files plus `my_directory`, which is highlighted with a red rectangle. Finally, `rmdir my_directory/` is executed, and a third `ls` command shows that `my_directory` has been removed. The terminal window has tabs for `PROBLEMS`, `DEBUG CONSOLE`, `TERMINAL` (selected), and `PORTS`. The status bar at the bottom shows `SSH: 143.248.188.31` and connection statistics.

```
PROBLEMS  DEBUG CONSOLE  TERMINAL  PORTS
cs230_user@taurus16:~$ ls
directory1 directory2 directory3 example file1 file2 file3
cs230_user@taurus16:~$ mkdir my_directory
cs230_user@taurus16:~$ ls
directory1 directory3 file1 file3
directory2 example file2 my_directory
cs230_user@taurus16:~$ rmdir my_directory/
cs230_user@taurus16:~$ ls
directory1 directory2 directory3 example file1 file2 file3
cs230_user@taurus16:~$
```

SSH: 143.248.188.31 0 0 0

Linux Commands

- `cat`
 - concatenate files and print on the standard output
 - `>` : redirects standard output to a file (overwrite)
 - `>>` : redirects standard output to a file (append)

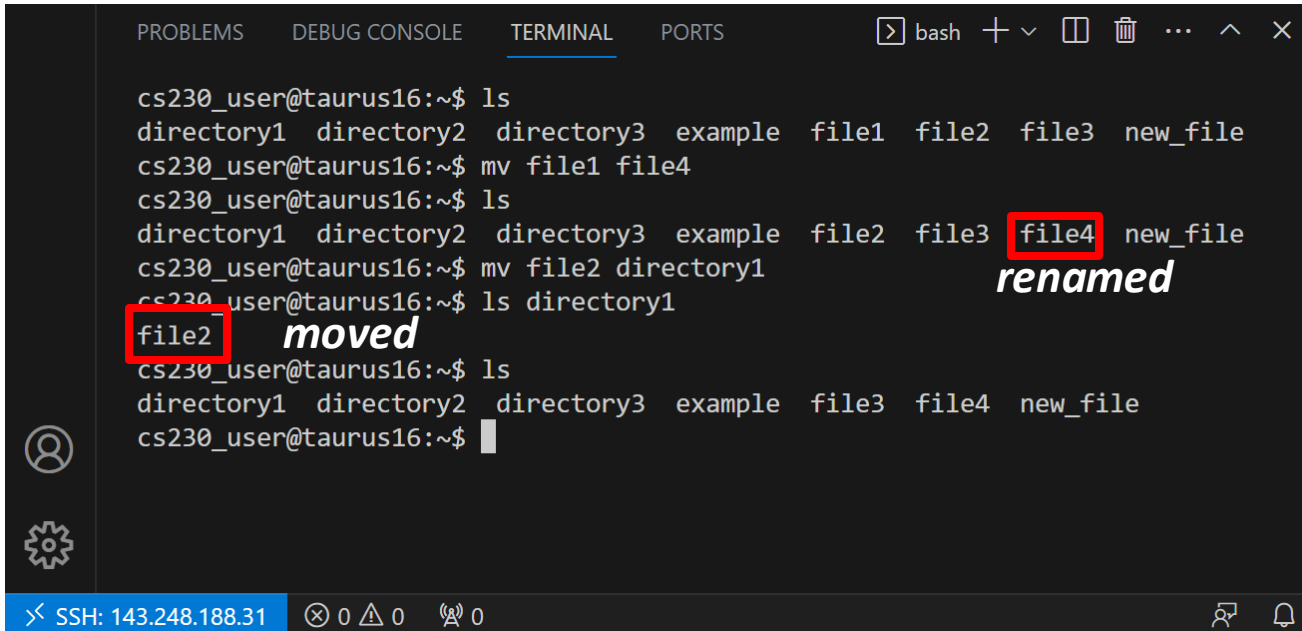


The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. To the right of the tabs are icons for a terminal window, a plus sign, a dropdown arrow, a window icon, a trash icon, and a menu icon. The terminal content shows a user named 'cs230_user' on a machine named 'taurus16' at the home directory. The user enters several `cat` commands: `cat file1`, `cat file2`, `cat file1 file2`, and `cat file1 file2 > new_file`. Each command is followed by its output, which is the concatenation of the contents of the specified files. The terminal also shows the user entering `cat new_file` to verify the output of the redirection. At the bottom of the terminal, there is a status bar with a blue background on the left showing 'SSH: 143.248.188.31', and on the right, icons for a terminal window, a plus sign, a dropdown arrow, a window icon, a trash icon, and a menu icon.

```
cs230_user@taurus16:~$ cat file1
Contents of file1
cs230_user@taurus16:~$ cat file2
Contents of file2
cs230_user@taurus16:~$ cat file1 file2
Contents of file1
Contents of file2
cs230_user@taurus16:~$ cat file1 file2 > new_file
cs230_user@taurus16:~$ cat new_file
Contents of file1
Contents of file2
cs230_user@taurus16:~$
```

Linux Commands

- `mv [source] [destination]`
 - Move or rename files or directories



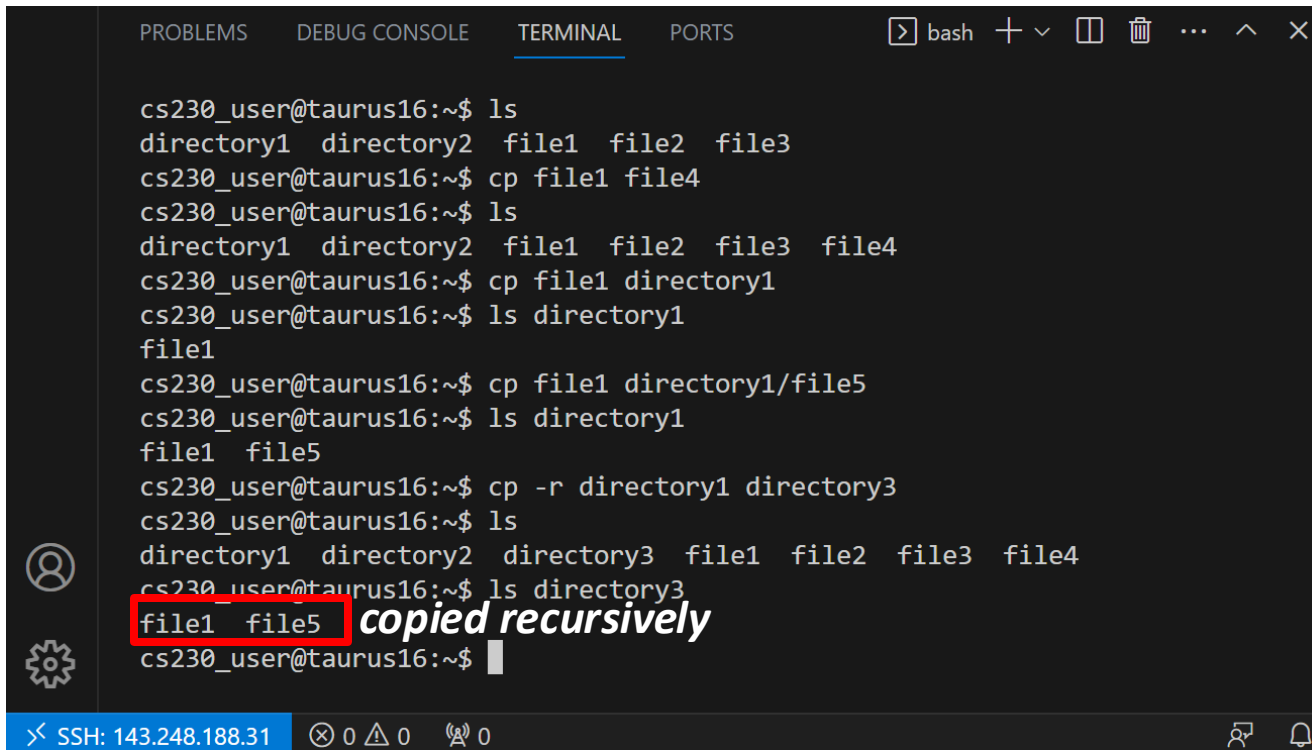
A terminal window showing a series of commands and their outputs. The terminal has a dark background with a light-colored text. The top bar shows tabs for PROBLEMS, DEBUG CONSOLE, TERMINAL (selected), and PORTS. The terminal content shows the user 'cs230_user' on host 'taurus16' performing several file operations. Red boxes highlight 'file4' and 'file2' in the output, with the word 'renamed' and 'moved' written next to them respectively. The bottom status bar shows 'SSH: 143.248.188.31' and connection statistics.

```
cs230_user@taurus16:~$ ls
directory1 directory2 directory3 example file1 file2 file3 new_file
cs230_user@taurus16:~$ mv file1 file4
cs230_user@taurus16:~$ ls
directory1 directory2 directory3 example file2 file3 file4 new_file
cs230_user@taurus16:~$ mv file2 directory1
cs230_user@taurus16:~$ ls directory1
file2
cs230_user@taurus16:~$ ls
directory1 directory2 directory3 example file3 file4 new_file
cs230_user@taurus16:~$
```

SSH: 143.248.188.31 0 0 0

Linux Commands

- `cp [source] [destination]`
 - Copy one or more files to specified location
 - `-r` : copy directories recursively



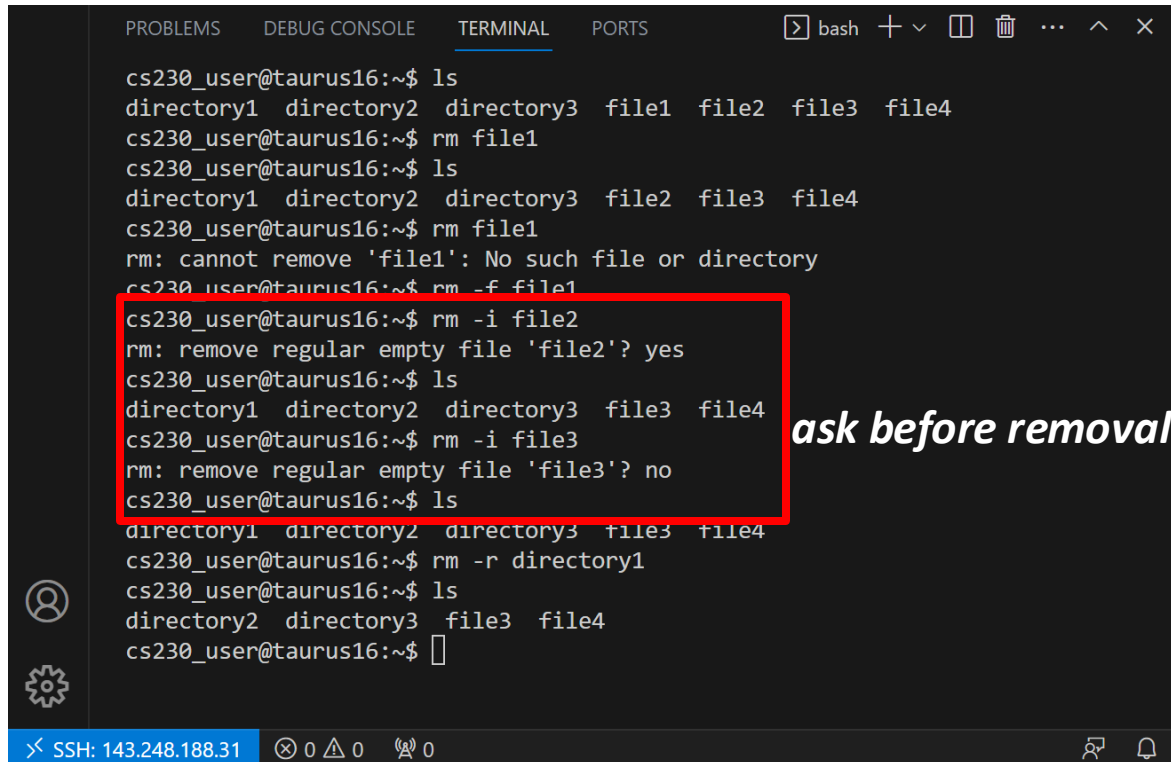
```
PROBLEMS  DEBUG CONSOLE  TERMINAL  PORTS  bash + - [ ] [ ] ... ^ X

cs230_user@taurus16:~$ ls
directory1 directory2 file1 file2 file3
cs230_user@taurus16:~$ cp file1 file4
cs230_user@taurus16:~$ ls
directory1 directory2 file1 file2 file3 file4
cs230_user@taurus16:~$ cp file1 directory1
cs230_user@taurus16:~$ ls directory1
file1
cs230_user@taurus16:~$ cp file1 directory1/file5
cs230_user@taurus16:~$ ls directory1
file1 file5
cs230_user@taurus16:~$ cp -r directory1 directory3
cs230_user@taurus16:~$ ls
directory1 directory2 directory3 file1 file2 file3 file4
cs230_user@taurus16:~$ ls directory3
file1 file5 copied recursively
cs230_user@taurus16:~$
```

SSH: 143.248.188.31 0 0 0

Linux Commands

- `rm [file/directory]`
 - remove files or directories
 - `-f` : ignore nonexistent files, never ask
 - `-i` : ask whether really remove file or not
 - `-r` : Remove directories and their contents recursively



```
PROBLEMS  DEBUG CONSOLE  TERMINAL  PORTS  bash + - [ ] [X] ... ^ X

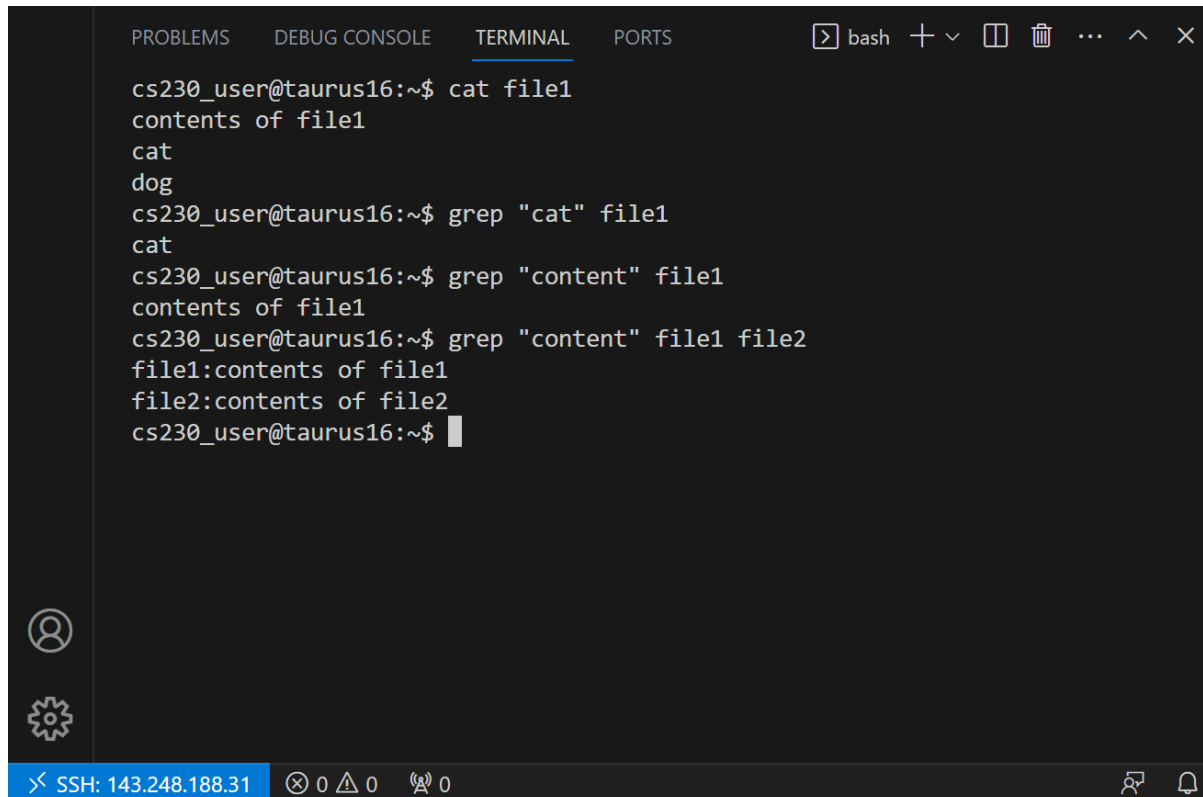
cs230_user@taurus16:~$ ls
directory1 directory2 directory3 file1 file2 file3 file4
cs230_user@taurus16:~$ rm file1
cs230_user@taurus16:~$ ls
directory1 directory2 directory3 file2 file3 file4
cs230_user@taurus16:~$ rm file1
rm: cannot remove 'file1': No such file or directory
cs230_user@taurus16:~$ rm -f file1
cs230_user@taurus16:~$ rm -i file2
rm: remove regular empty file 'file2'? yes
cs230_user@taurus16:~$ ls
directory1 directory2 directory3 file3 file4
cs230_user@taurus16:~$ rm -i file3
rm: remove regular empty file 'file3'? no
cs230_user@taurus16:~$ ls
directory1 directory2 directory3 file3 file4
cs230_user@taurus16:~$ rm -r directory1
cs230_user@taurus16:~$ ls
directory2 directory3 file3 file4
cs230_user@taurus16:~$
```

ask before removal

SSH: 143.248.188.31 0 0 0 0

Linux Commands

- `grep [pattern] [files]`
 - print lines matching a pattern



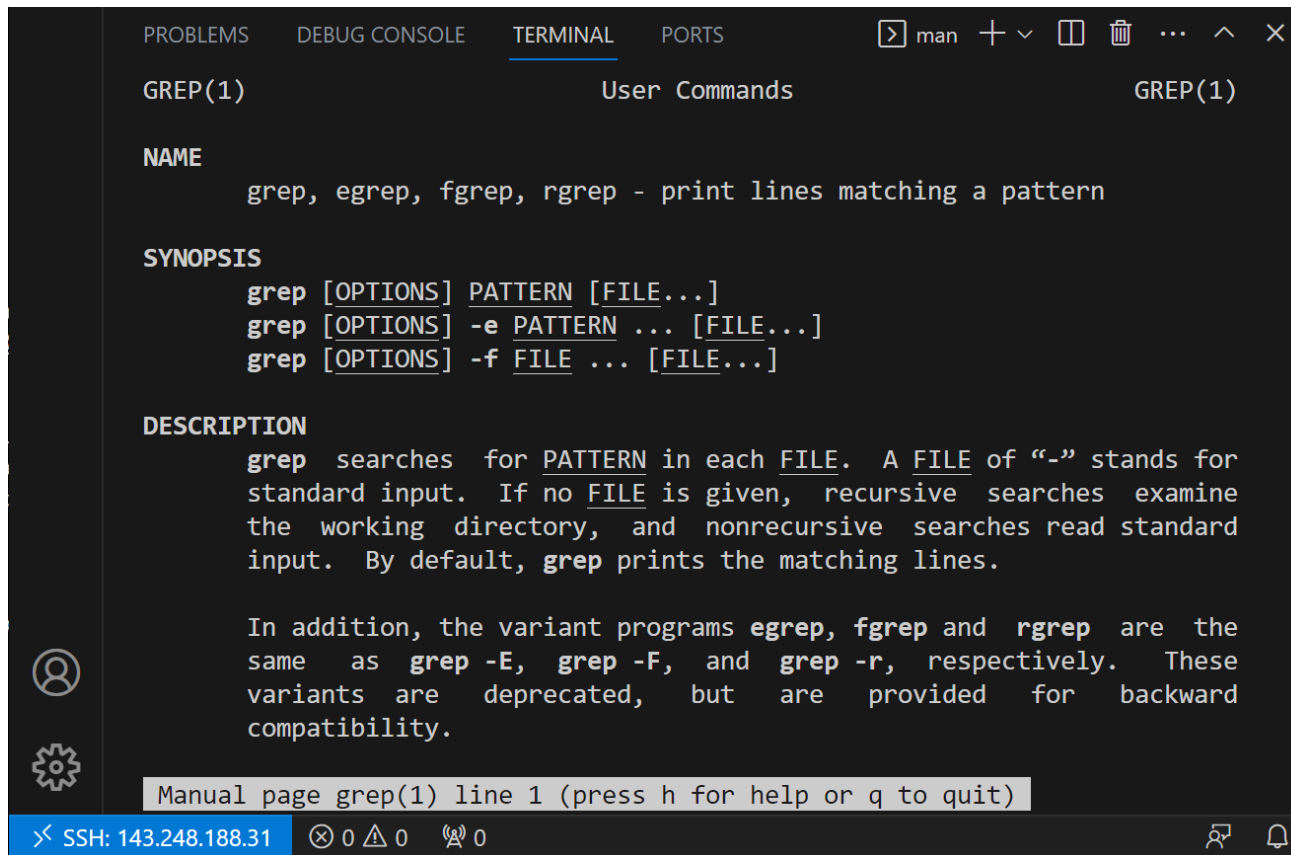
The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. The terminal displays the following commands and their outputs:

```
cs230_user@taurus16:~$ cat file1
contents of file1
cat
dog
cs230_user@taurus16:~$ grep "cat" file1
cat
cs230_user@taurus16:~$ grep "content" file1
contents of file1
cs230_user@taurus16:~$ grep "content" file1 file2
file1:contents of file1
file2:contents of file2
cs230_user@taurus16:~$
```

At the bottom of the terminal, there is a status bar showing 'SSH: 143.248.188.31' and some system icons.

Linux Commands

- `man [command name]`
 - an interface to the on-line reference manuals



The screenshot shows a terminal window with a dark theme. The title bar at the top includes tabs for 'PROBLEMS', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. To the right of the tabs are icons for a search box (containing 'man'), a plus sign, a dropdown arrow, a window icon, a trash icon, an ellipsis, and up/down arrows. The terminal content displays the manual page for 'grep(1)', titled 'User Commands'. The page is structured with sections: 'NAME' (describing grep, egrep, fgrep, and rgrep), 'SYNOPSIS' (showing command-line options like -E, -e, -f), and 'DESCRIPTION' (explaining how grep searches for patterns in files). At the bottom of the terminal, a status bar shows 'SSH: 143.248.188.31' and connection statistics. On the left side of the terminal window, there are icons for a user profile and a settings gear.

```
PROBLEMS  DEBUG CONSOLE  TERMINAL  PORTS  > man + v [] [trash] ... ^ x
GREP(1)                                     User Commands                                GREP(1)

NAME
    grep, egrep, fgrep, rgrep - print lines matching a pattern

SYNOPSIS
    grep [OPTIONS] PATTERN [FILE...]
    grep [OPTIONS] -e PATTERN ... [FILE...]
    grep [OPTIONS] -f FILE ... [FILE...]

DESCRIPTION
    grep searches for PATTERN in each FILE. A FILE of "-" stands for
    standard input. If no FILE is given, recursive searches examine
    the working directory, and nonrecursive searches read standard
    input. By default, grep prints the matching lines.

    In addition, the variant programs egrep, fgrep and rgrep are the
    same as grep -E, grep -F, and grep -r, respectively. These
    variants are deprecated, but are provided for backward
    compatibility.

Manual page grep(1) line 1 (press h for help or q to quit)
x SSH: 143.248.188.31  x 0 A 0  A 0  [chat] [bell]
```


Linux Commands

- Running a command in background :
\$ [command] &
- Running many commands using a single line :
\$ [command1] ; [command2] ; [command3]
- Use the output of [command1] as an input of [command2] :
\$ [command1] | [command2]
- Save the printed output of command in a file :
\$ [command] > [file to save]

gcc

- gcc
 - gcc hello.c -E -o hello.i ← preprocessed file
 - gcc hello.c -S -o hello.s ← assembly file obj
 - gcc hello.c -c -o hello.o ← ect file

make

- make
 - Make is a utility that automatically builds executable programs and libraries from source code by reading files called Makefiles (Wikipedia)
- Structure of Makefiles

```
target ... : prerequisites ...  
recipe
```

```
...
```

```
...
```

make

- A sample Makefile

```
$cat Makefile
```

```
final: main.o end.o inter.o start.o
```

```
    gcc -o final main.o end.o inter.o start.o
```

```
main.o: main.c global.h
```

```
    gcc -c main.c
```

```
end.o: end.c local.h global.h
```

```
    gcc -c end.c
```

```
inter.o: inter.c global.h
```

```
    gcc -c inter.c
```

```
start.o: start.c global.h
```

```
    gcc -c start.c
```

```
clean:
```

```
    rm -f main.o end.o inter.o start.o
```

make

- A sample Makefile

```
$make  
$ls | grep final  
final
```

Self-study material

- Linux : <https://www.youtube.com/watch?v=CpTfQ-q6MPU>