

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

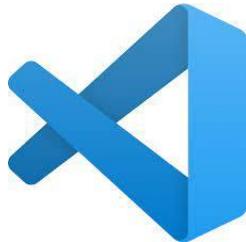
- How can I use a Linux machine?
 - You have to establish a connection to it.
- How can I connect to a machine?
 - Use ssh clients like PuTTY or **VSCode**.



PuTTY

PuTTY: A Free Telnet/SSH Client

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



VSCode

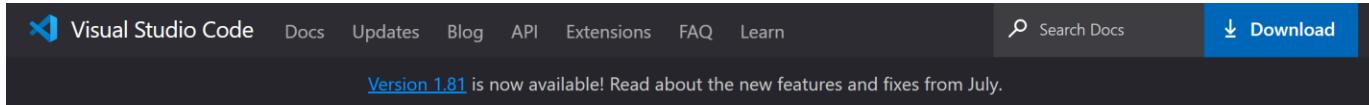
VSCode (Visual Studio Code)

- 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**



↓ Windows

Windows 10, 11



↓ .deb

Debian, Ubuntu

↓ .rpm

Red Hat, Fedora, SUSE



↓ Mac

macOS 10.11+

User Installer x64 x86 Arm64

System Installer x64 x86 Arm64

.zip x64 x86 Arm64

CLI x64 x86 Arm64

.deb x64 Arm32 Arm64

.rpm x64 Arm32 Arm64

.tar.gz x64 Arm32 Arm64

Snap Snap Store

CLI x64 Arm32 Arm64

.zip Intel chip Apple silicon Universal

CLI Intel chip Apple silicon

**Choose it if you are
using Windows**

Connection Example: VSCode

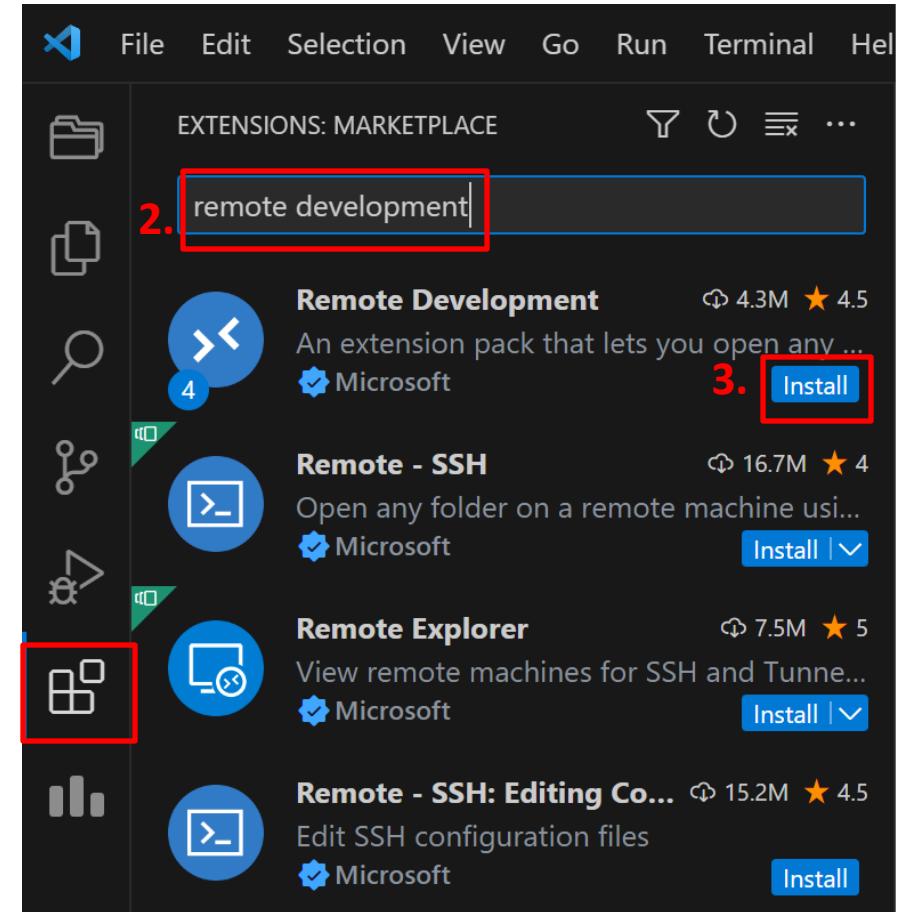
Assuming you have already installed VSCode in your local machine

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

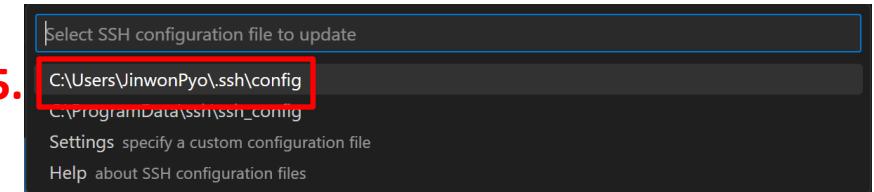
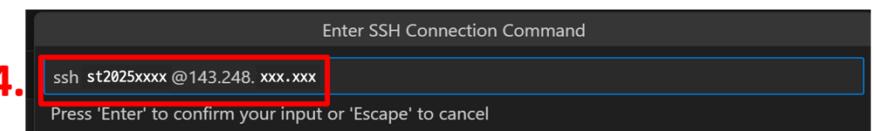
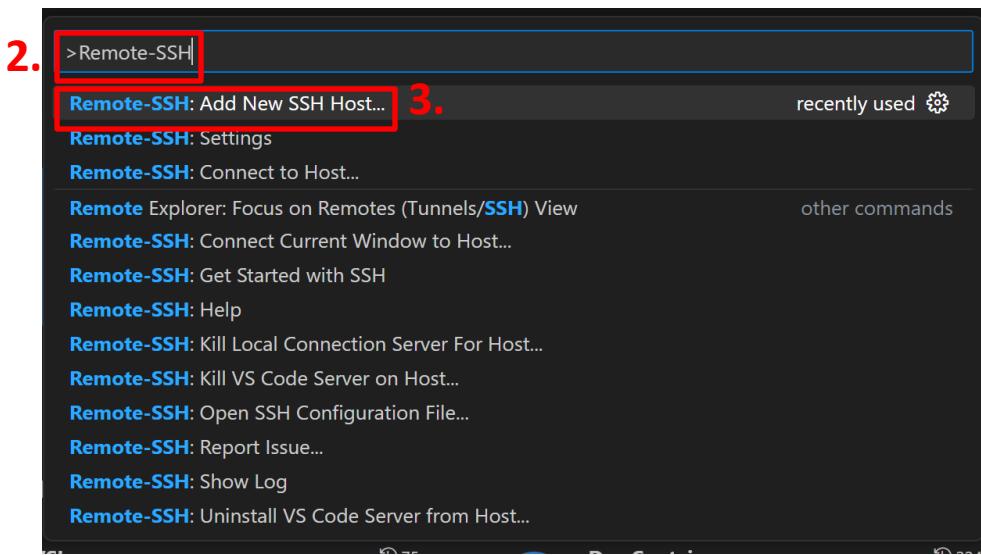
IP address: Your assigned IP address

ID: st[your student id]

Password: Your assigned password

Steps.

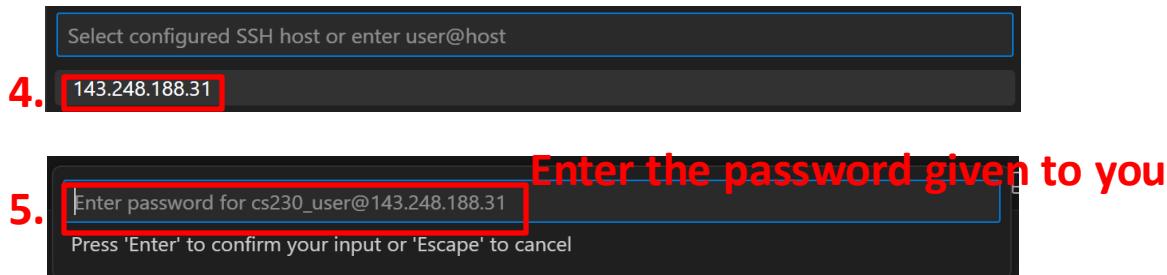
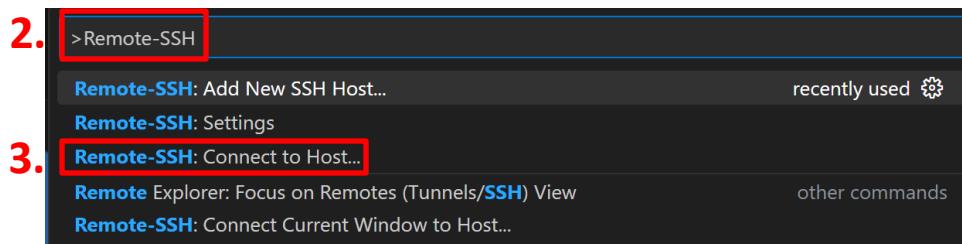
1. Hit the “F1” key (dropdown menu will pop)
2. Search “Remote-SSH”
3. Select “Add New SSH Host...”
4. Enter the given information as the following format (**When asked to choose the targeting OS, then choose “Linux”**)
5. Just select the first option



Connection Example: VSCode - 3

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

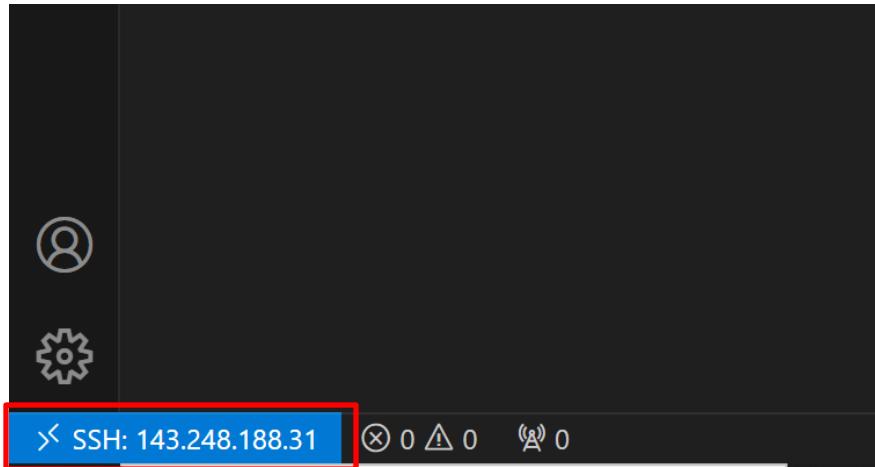
1. Hit the “F1” key (dropdown menu will pop)
2. Search “Remote-SSH”
3. Select “Connect to Host...”
4. Select the IP address you entered during the previous step
5. New VSCode window will pop out and ask you to enter password



Connection Example: VSCode - 4

You are now in the remote machine!

Check out the bottom left corner of your VSCode window

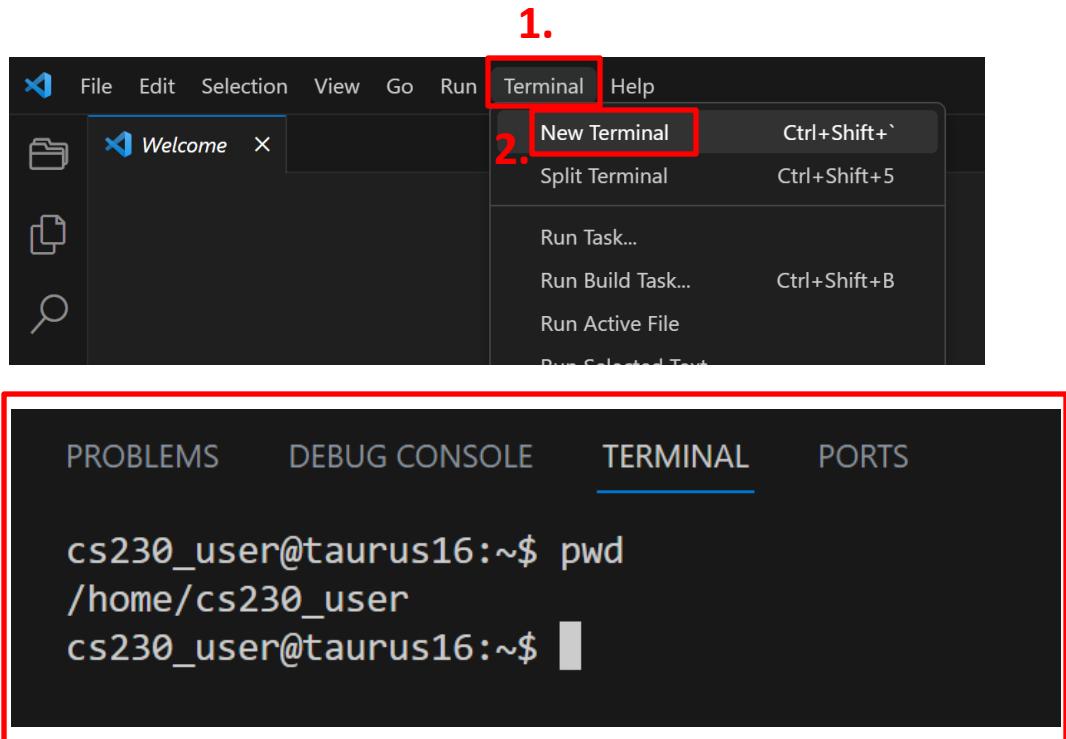


Connection Example: VSCode - 5

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

Steps.

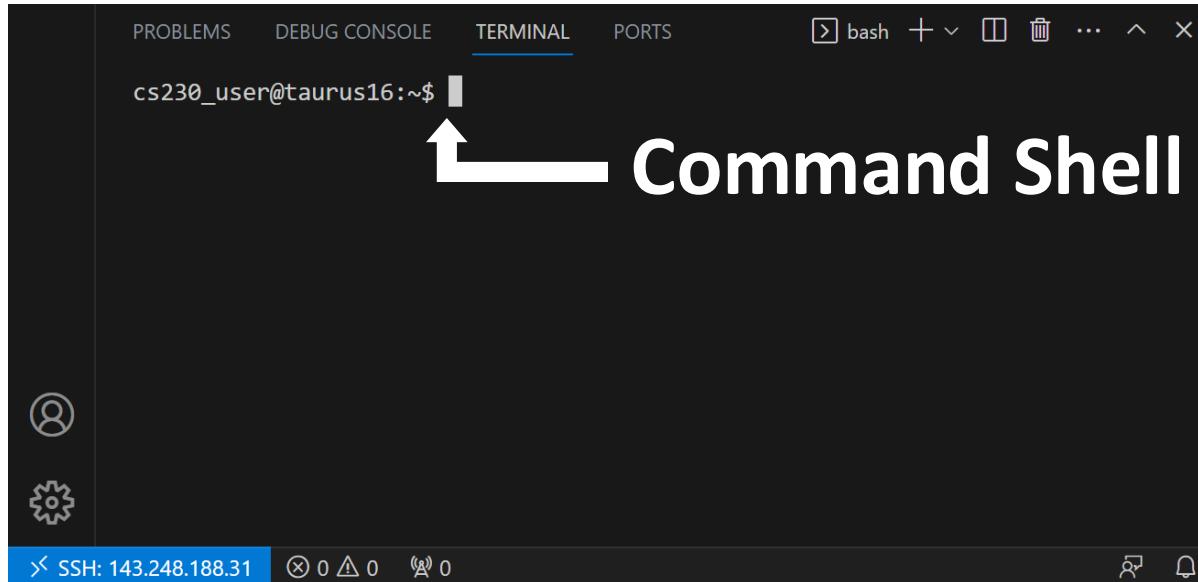
1. Find “Terminal”
2. Select “New Terminal”
3. Then you will see the Linux shell (or terminal) pop up in the bottom of VSCode window



3. This is the Linux shell

Linux Shell

Connected!



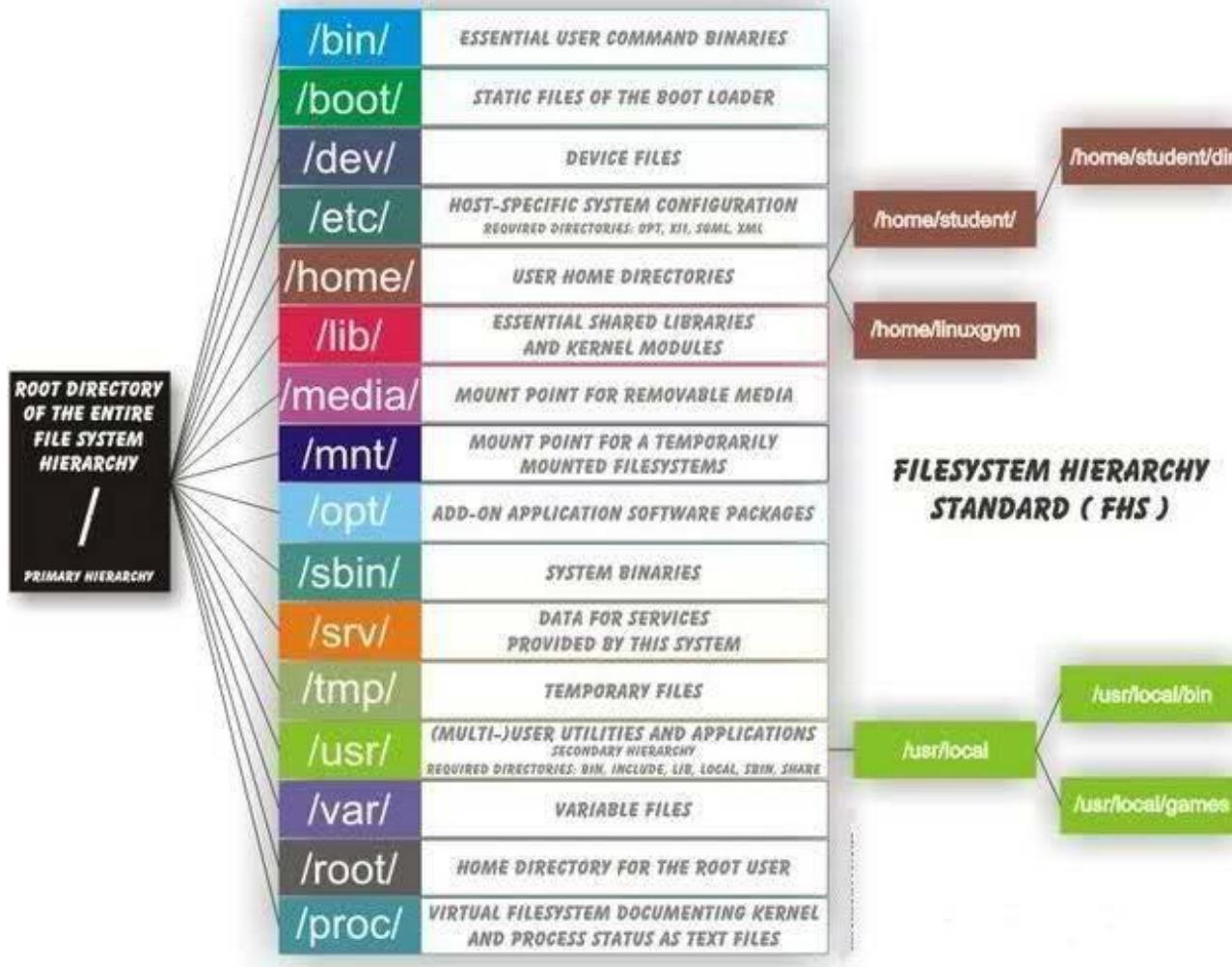
Shell?

- a command line interface between a user and Linux machine

Linux Directory

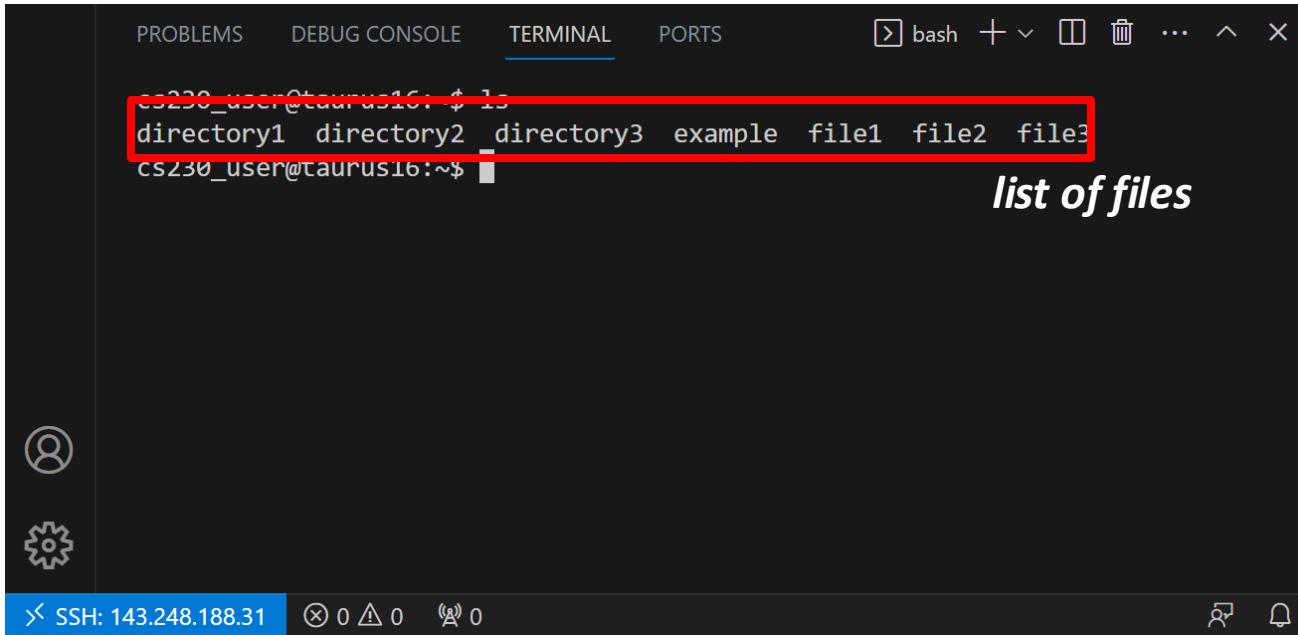
- / : root directory
- ~ : user's home directory (usually same as /home/[username])
- . : current(working) directory
- .. : upper(parent) directory

Linux Directory Structure



Linux Commands

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



A screenshot of a terminal window in the Visual Studio Code interface. The window title is "TERMINAL". The status bar at the bottom shows "SSH: 143.248.188.31". The terminal content is as follows:

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

The output of the ls command is highlighted with a red rectangle. To the right of the highlighted text, the words "list of files" are written in a italicized font.

Linux Commands

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

```
PROBLEMS DEBUG CONSOLE TERMINAL PORTS bash + ▾ □ ⚡ ... ^ ×  
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 ⑧ 0 Δ 0 ⑨ 0 ⚡ 🔔

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

cs230_user@taurus16:~\$ ls -l

total 16

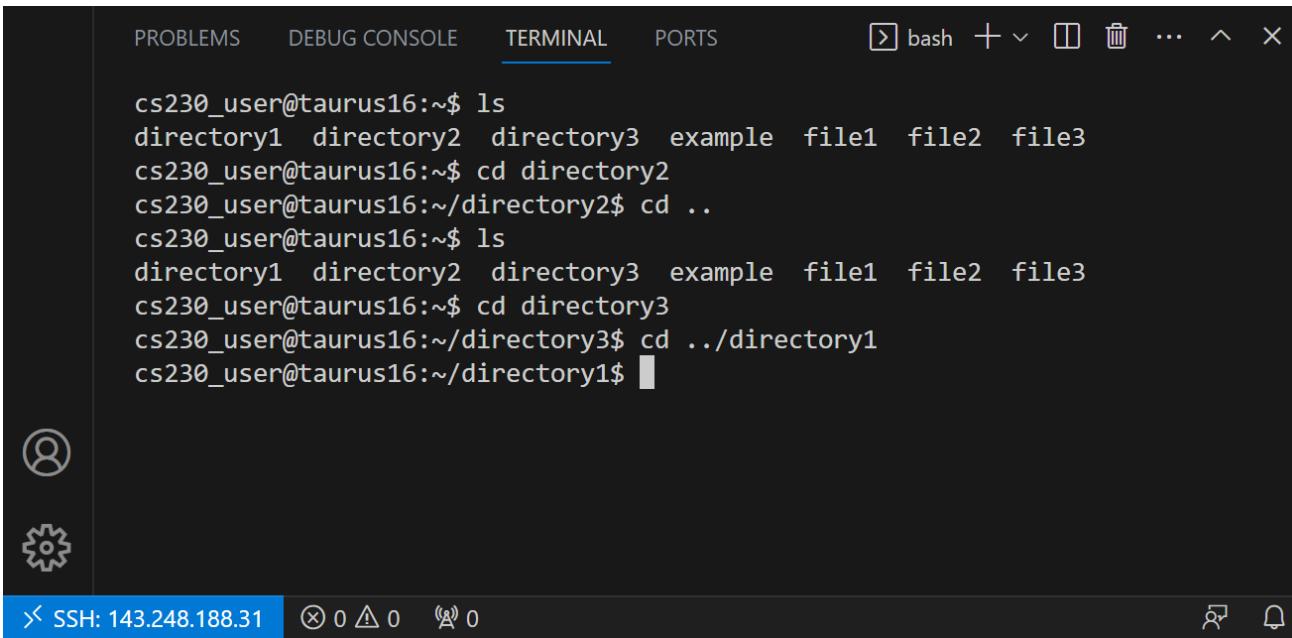
```
drwxrwxr-x 2 cs230_user cs230_user 4096 Aug 25 17:09 directory1
drwxrwxr-x 2 cs230_user cs230_user 4096 Aug 25 17:09 directory2
drwxrwxr-x 2 cs230_user cs230_user 4096 Aug 25 17:09 directory3
drwxrwxr-x 2 cs230_user cs230_user 4096 Aug 25 17:26 example
-rw-rw-r-- 1 cs230_user cs230_user    0 Aug 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

Linux Commands

- cd [directory]
 - change working directory



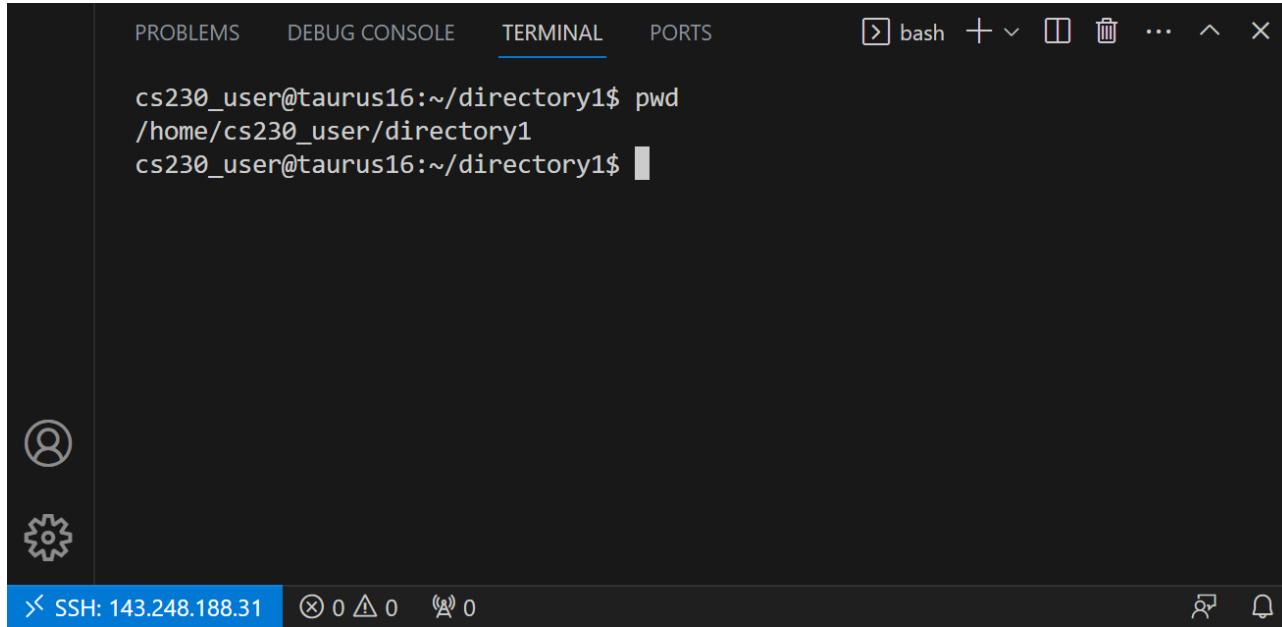
The screenshot shows a terminal window in the Visual Studio Code interface. The terminal tab is active, displaying a session on a remote host (SSH: 143.248.188.31). The user has run several commands to demonstrate directory navigation:

```
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$ █
```

The terminal interface includes standard VS Code icons for problems, debug console, ports, and a gear for settings. The status bar at the bottom shows the connection details and some activity counts.

Linux Commands

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



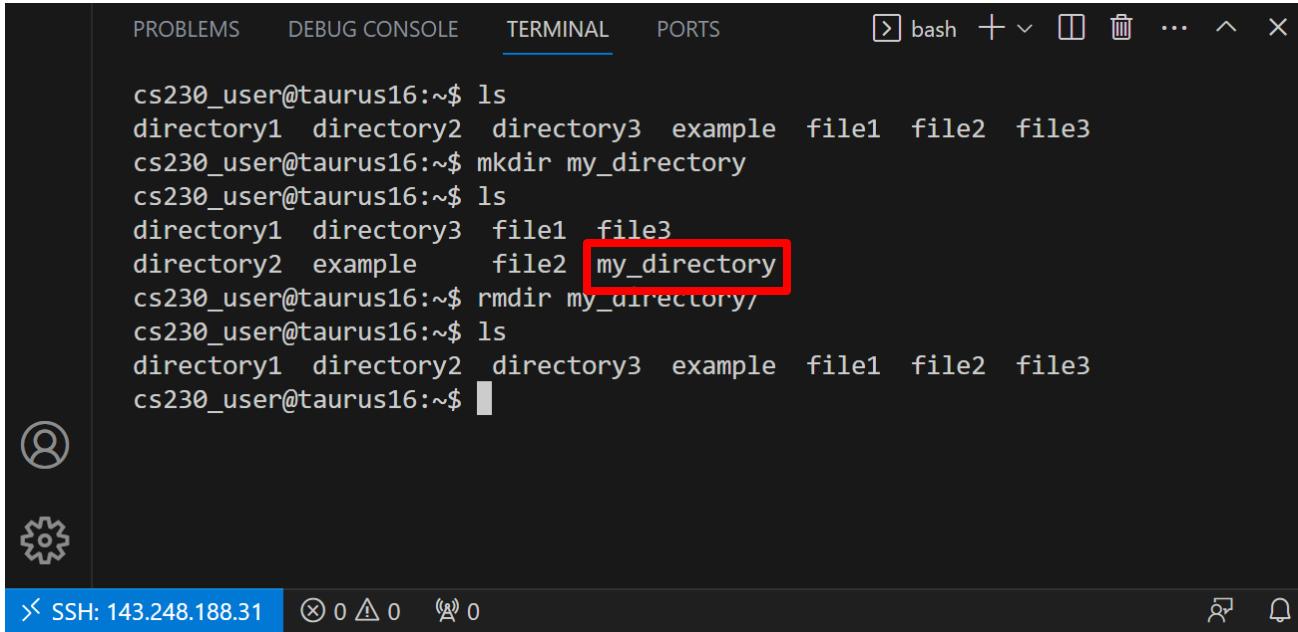
A screenshot of a terminal window in the Visual Studio Code interface. The window title is "bash". The terminal tab is selected, showing the following text:

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

The terminal window has a dark background. On the left side, there is a vertical sidebar with icons for a user profile and a gear. At the bottom of the terminal window, there is a status bar displaying "SSH: 143.248.188.31" and some status indicators.

Linux Commands

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



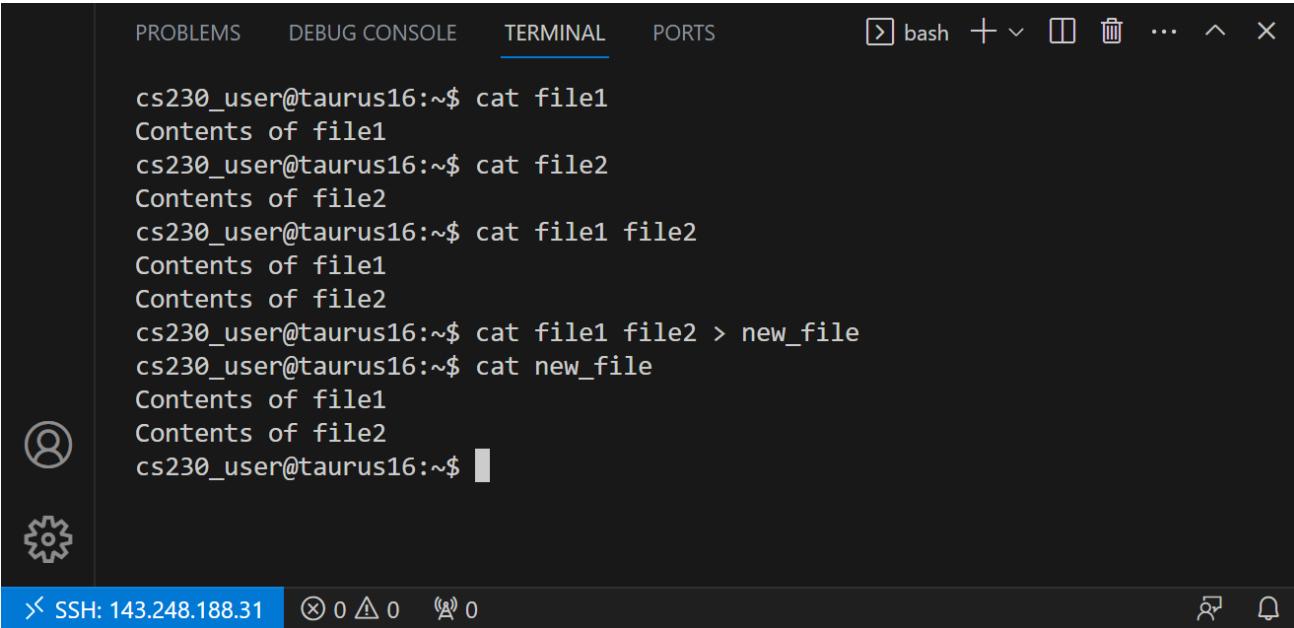
The screenshot shows a terminal window in the VS Code interface. The terminal tab is selected, showing the following command history:

```
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:~$
```

The directory name "my_directory" is highlighted with a red box.

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)

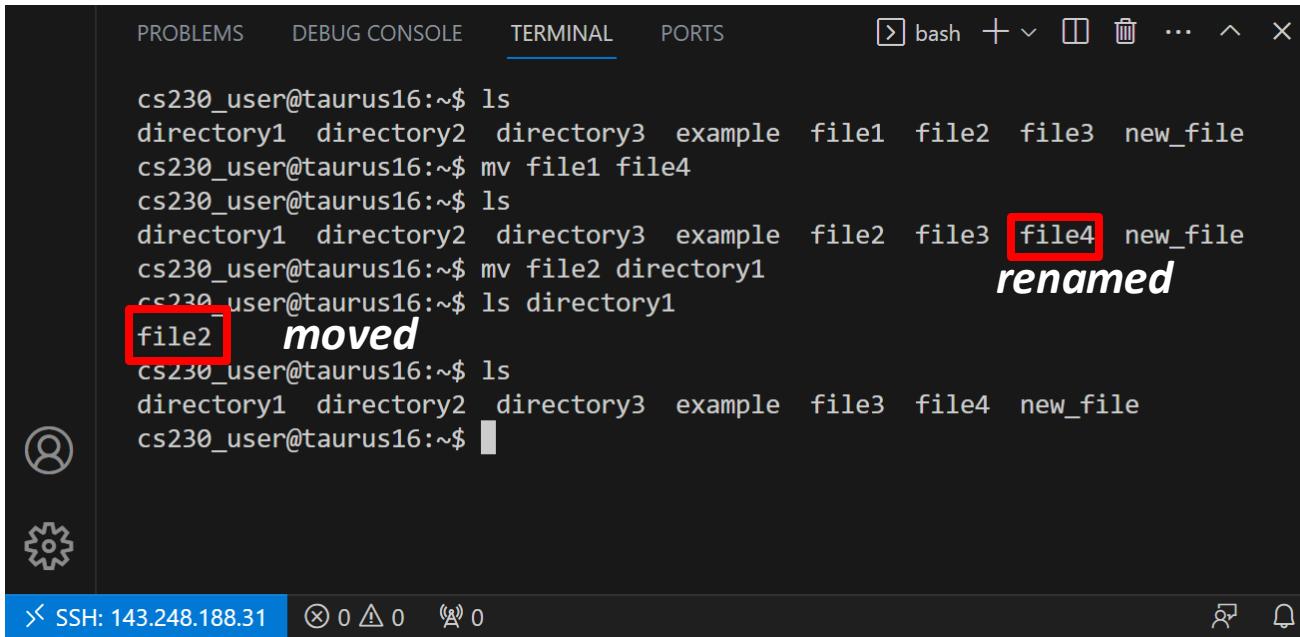


The screenshot shows a terminal window in the Visual Studio Code interface. The terminal tab is selected, displaying a series of Linux commands and their outputs. The user, cs230_user, is running several 'cat' commands to display the contents of 'file1' and 'file2', and then concatenating them into a new file named 'new_file'. The terminal also shows the contents of 'file1' and 'file2' separately. The bottom status bar indicates an SSH connection to 143.248.188.31.

```
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



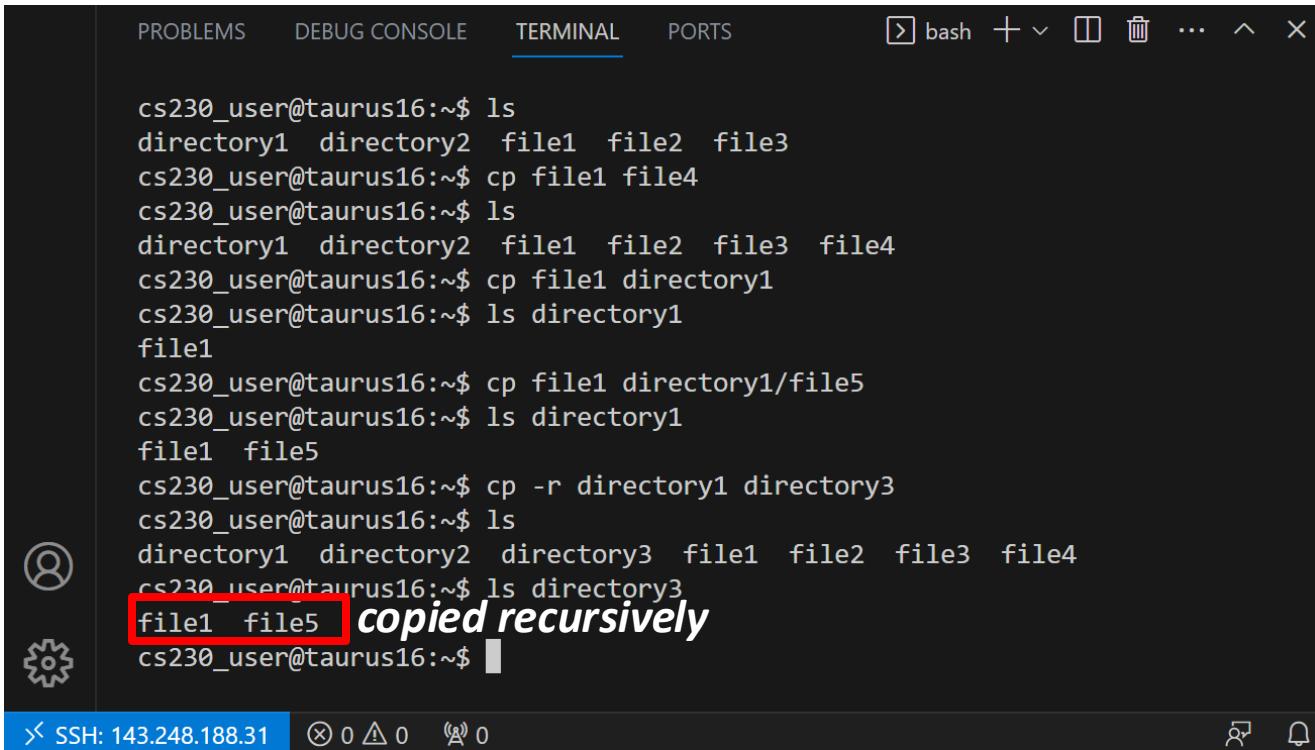
The screenshot shows a terminal window with the following session:

```
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 moved
cs230_user@taurus16:~$ ls
directory1 directory2 directory3 example file3 file4 new_file
cs230_user@taurus16:~$
```

In the terminal output, the file "file2" is highlighted with a red box and labeled "moved". The file "file4" is also highlighted with a red box and labeled "renamed".

Linux Commands

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



The screenshot shows a terminal window with the following session:

```
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:~$
```

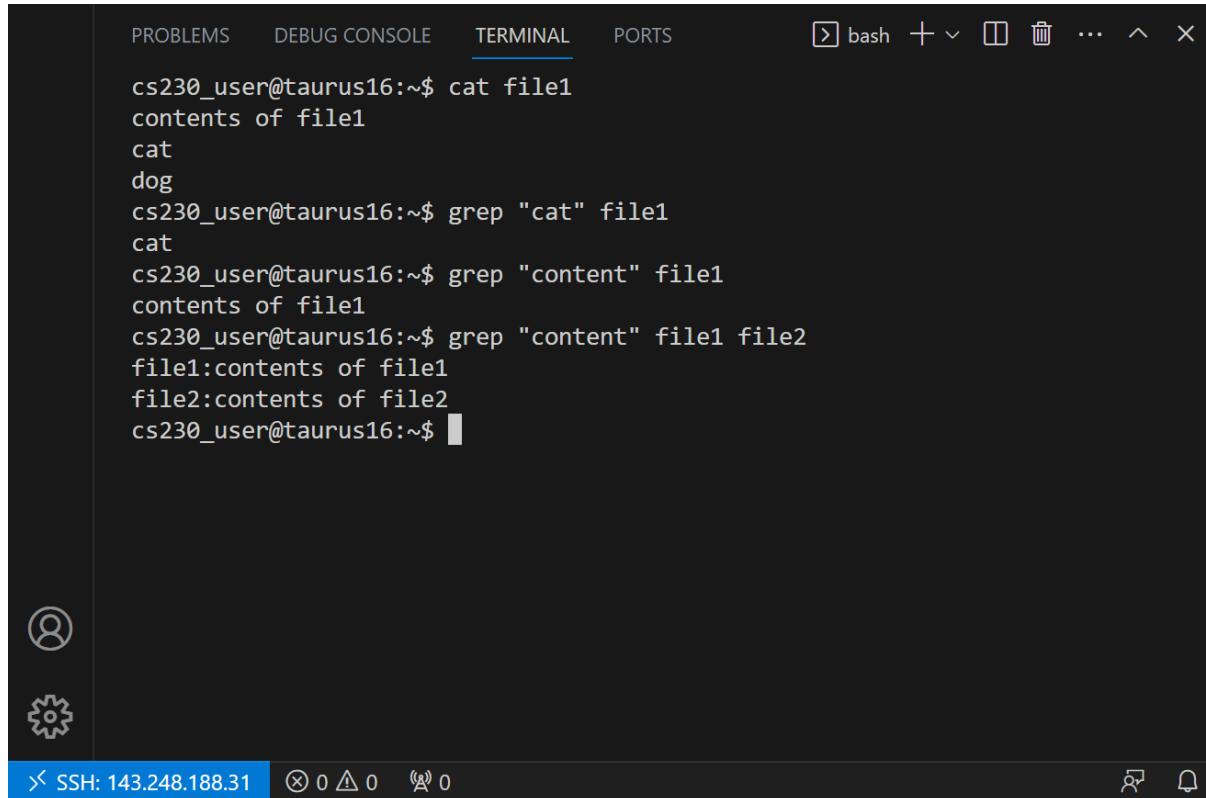
The line "copied recursively" is highlighted with a red box.

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

Linux Commands

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



The screenshot shows a terminal window in the VS Code interface. The terminal tab is selected, and the title bar shows "bash". The command history and output are as follows:

```
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:~$
```

The bottom status bar indicates the connection is via SSH to 143.248.188.31.

Linux Commands

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

```
PROBLEMS DEBUG CONSOLE TERMINAL PORTS
Σ man + ~ ⌂ ⌂ ... ^ ×
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)
× SSH: 143.248.188.31 ⌂ 0 △ 0 ⌂ 0 ⌂ ⌂
```

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 -c end.c
inter.o: inter.c global.h
        gcc c -c inter.c s
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>