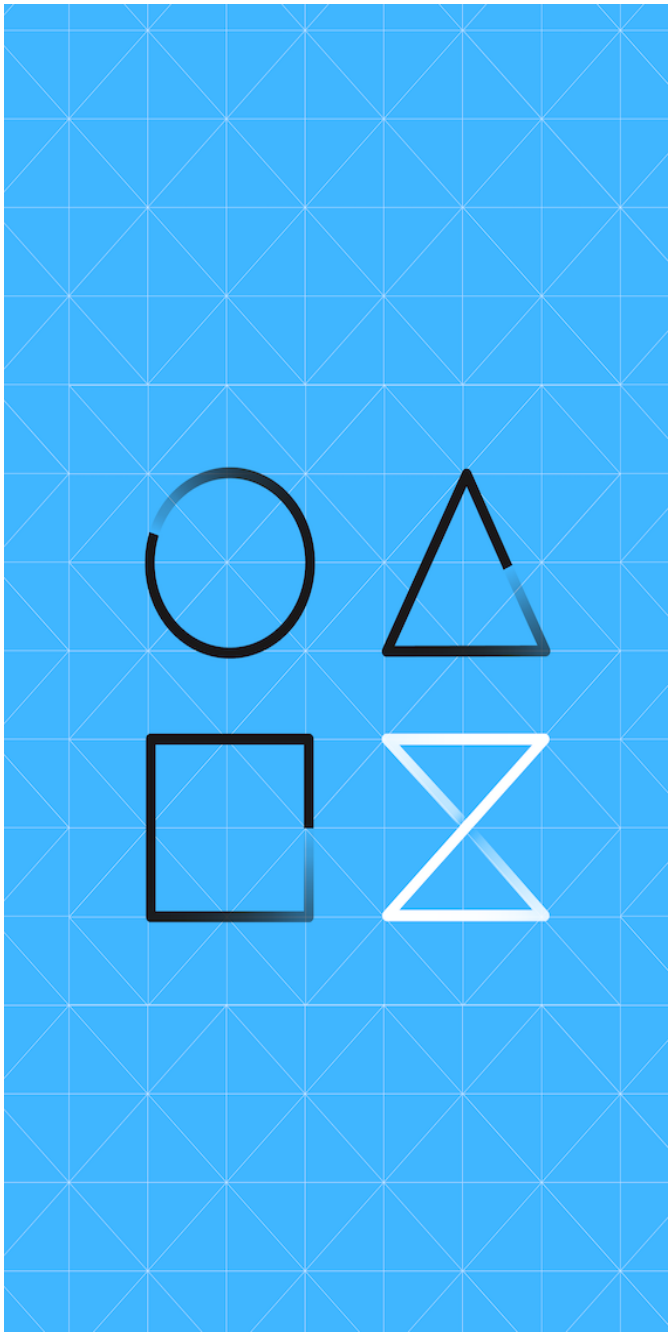


Fundamentals

250417-2338

USS1

Let’s Dive Deeper

- CD CIRCLE && CD .. && CD - && ./SHAKE_IT_ALL_ABOUT.SH
- 1 FIND THE TERMINAL
- 2 CONNECT THROUGH SSH
- 3 VERIFY ME, THANK YOU
- 4 YOU’RE IN! NOW WHAT?
- 5 GETTING ORIENTATED
- 6 CHANGE DIRECTORY WITH CD
- 7 FRESH FILES AND FOLDERS
- 8 INVITE ZOWE TO THE PARTY
- 9 I’VE GOT A SECRET
- 10 REDIRECTING THE OUTPUT
- 11 SPACE EXPLORATION
- 12 MAKE IT COUNT

250417-2338

CD CIRCLE && CD .. && CD - && ./SHAKE_IT_ALL_ABOUT.SH

The Challenge

UNIX System Services (USS) is a POSIX-compliant implementation of a UNIX environment within z/OS that allows for a UNIX-like experience while still using the same system APIs as the z/OS you have been using so far.

All that will make more sense as you make your way through the shell, run shells scripts, and fall hopelessly in love with tab completion.

Before You Begin

We're starting with a brand-new facet of z/OS, so you don't really need to know much else, but you will need your VSCode environment.

A knowledge of how data sets and members work will certainly help put things into perspective as you learn about USS.

Investment

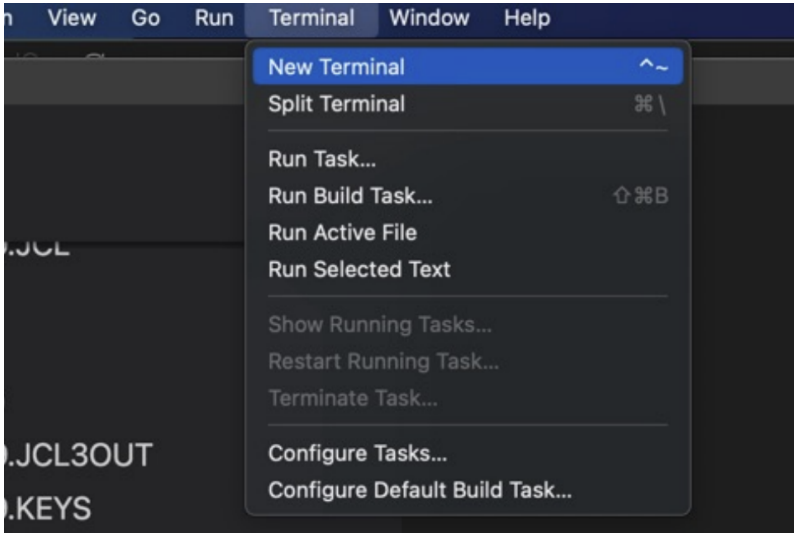
Steps	Duration
12	60 minutes

USS1250417-2338

1 FIND THE TERMINAL

Look for the Terminal section in the lower portion of your VSCode window.

If you don’t see it, try using the Terminal menu option and selecting “New Terminal”.



This is a text-based method of interacting with your own personal system.

If you have an older version of Windows, you may need to download a separate SSH client, such as PuTTY.

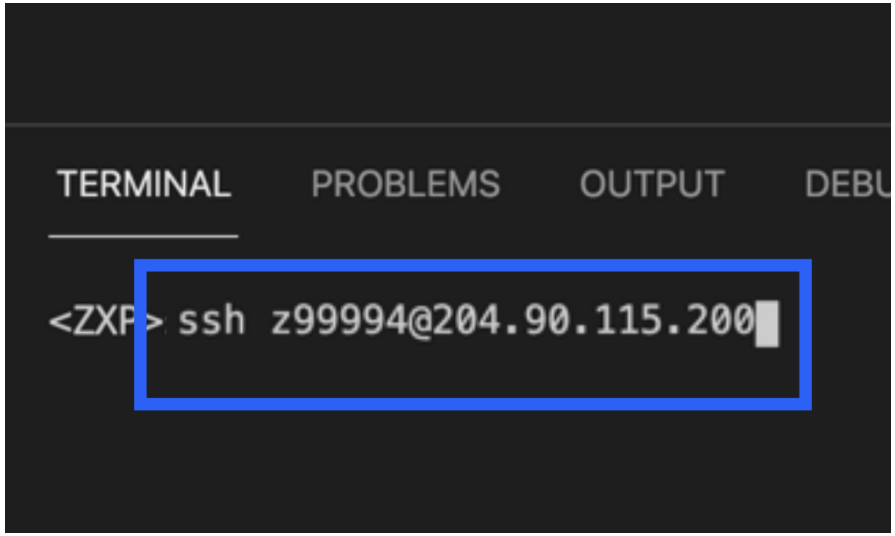
For guidance on this, consult the support forums.

USS1250417-2338

2 CONNECT THROUGH SSH

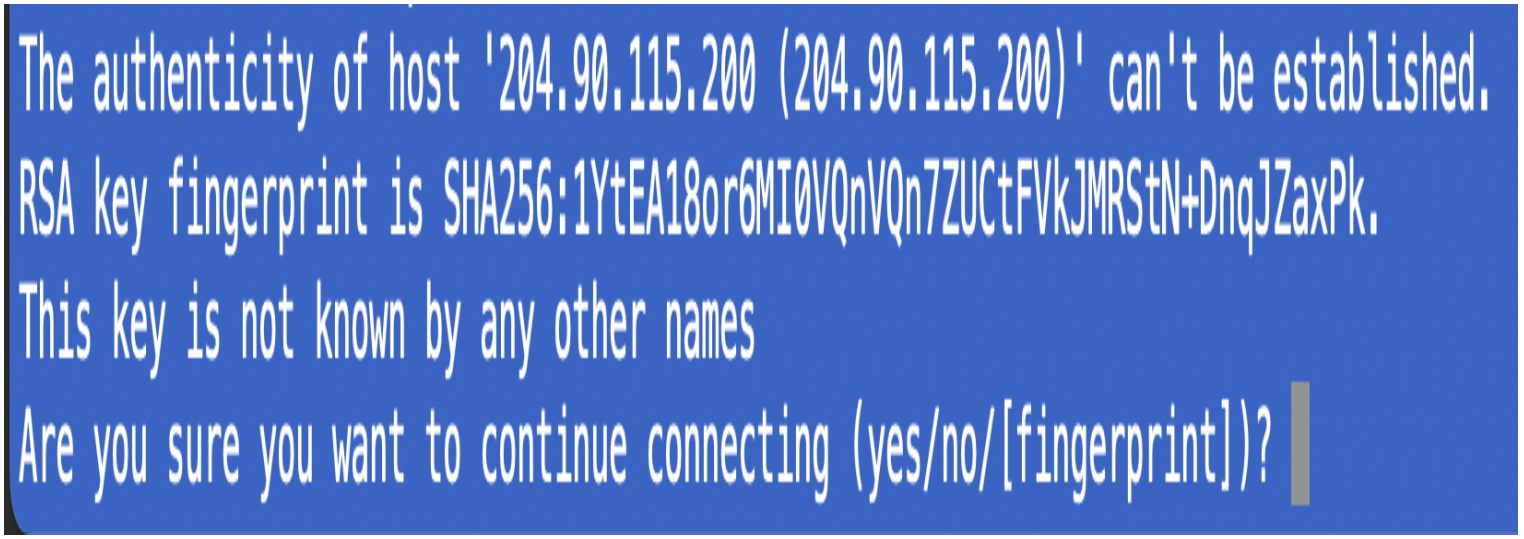
Log into the z/OS system with the following command:

ssh zxxxxx@204.90.115.200 (replacing zxxxxx with your own userid).



To put another way, this command says “Use the ssh command to connect me (using my userid) to the remote system (at the target IP address)”

The first time you connect, you may see a message about approving continued connection:



This is normal - answer “yes”, and continue.

3 VERIFY ME, THANK YOU

You will now be asked for your password, which is the same password you used to log into the z/OS system through VSCode.

A few things to note:

- You may see “Authenticity of host ... can’t be established”
- You may be prompted to trust or accept a key from the remote system.

You can safely answer “yes” to any of these prompts.

You will not see any characters as you type your password!

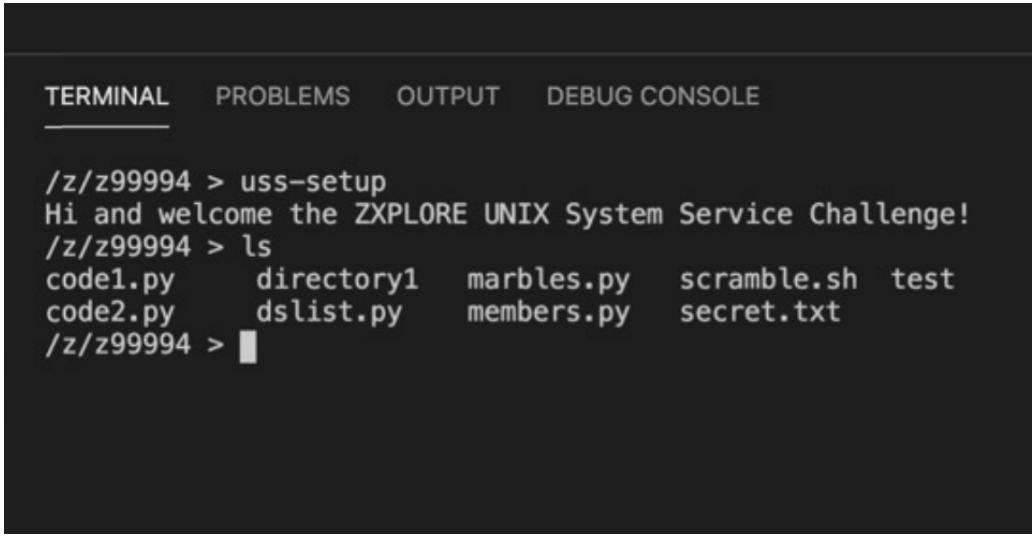
This is to make it so people looking over your shoulder can’t steal your password, but the system can still see it.

4 YOU'RE IN! NOW WHAT?

Now that you're logged into the USS environment with ssh, you can look around with the command ls.

This will show all of the files and folders (also known as directories in UNIX-land).

Your default ("home") directory will be empty - enter the command uss-setup, and then run the ls command again.



USS

USS1250417-2338

USS uses a hierarchical structure, where there are files and directories within other directories.

You may be used to this type of file-system on your own computer, where you can keep files in folders, and put folders within folders and so on.

When you want to disconnect, just enter the command exit and you will be logged out. You can use the ssh command from your laptop to get reconnected later.

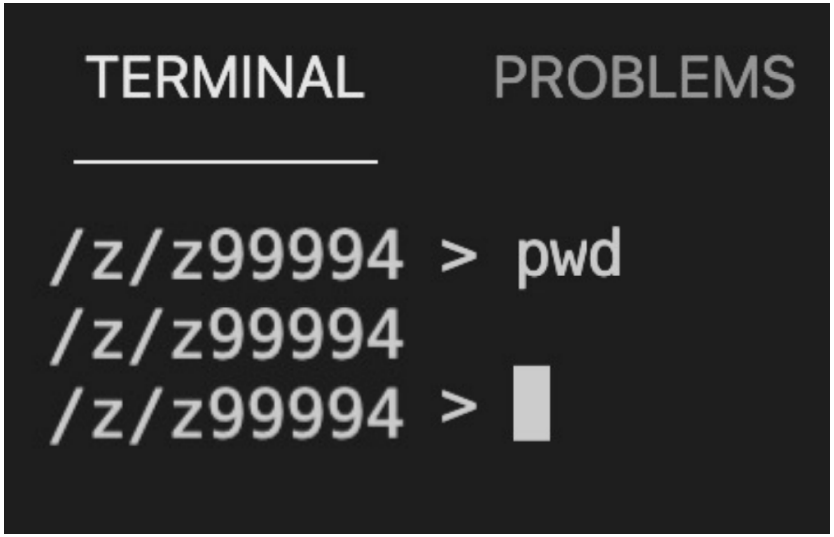
If you leave the ssh session idle for more than a few minutes (usually 3-5 minutes), it will become unresponsive. If that happens, your terminal will appear stuck.

To clear this, enter `~.` – you may need to do this more than once to get the session to disconnect.

5 GETTING ORIENTATED

In addition to looking around with `ls`, you may also want to know “Where am I?”.

The default command prompt will usually show you where you are in the directory structure, but you can also type `pwd` to *P*rint the *W*orking *D*irectory.



This will be useful shortly when you will start using `cd` to *C*hange *D*irectory to move around the filesystem.

Right now, you’re in your home directory, which is where *your* `USS` files live.

You can get back to this directory at any time by typing `cd ~` (that’s the tilde key)

6 CHANGE DIRECTORY WITH CD

To navigate into another directory, type `cd`, followed by the name of the directory.

For example, we can type `cd directory1` and we'll go into directory1, assuming that's a directory we can see with the `ls` command.

Try it out, then type `pwd`

You should see that pwd now shows your path as `/z/zxxxxx/directory1`

To go back to your home directory, you need to go back one level. You can do this by typing `cd ..` (two dots), or you can use the tilde shortcut to go straight to your home directory from anywhere (`cd ~`)

So far, you have been moving back and forth, one step at a time. You can also change directories by specifying the full path to the directory that you want to go to.

For example, `cd /z/public/test` will take you directly to that new location.

Then use `cd ~` to go back home, after we've gone there and looked around a little.

(Again, if your home directory is empty, run the uss-setup command)

MAKE LIFE EASIER. USE TAB COMPLETION AND UP/DOWN ARROWS

Notice how the word “directory” gets annoying to have to type out after a while? Try this neat trick ... the next time you have to type out a long command or filename, type out the first few letters, then press the Tab key.

Hitting it once will auto-complete as much of the command or name as it can, and hitting it twice will show you all of the possible completions. So, if you have a “directory1” and a “directory2”, you can type `cd di` followed by pressing the Tab key, and it will auto-complete the “directory” part.

Press it one more time and it will tell you “there’s directory1 and directory 2”.

Give it a try, and you’ll be amazed at how much faster you can get with the command prompt.

One more trick, if you want to use a command you typed not that long ago, you can recall recent commands by pressing the Up arrow on your keyboard. Then just hit Enter to use it. Pretty sweet, right?

If you need to alter a previous command, use the Backspace key to remove unwanted characters, and then type what you need.

USS1250417-2338

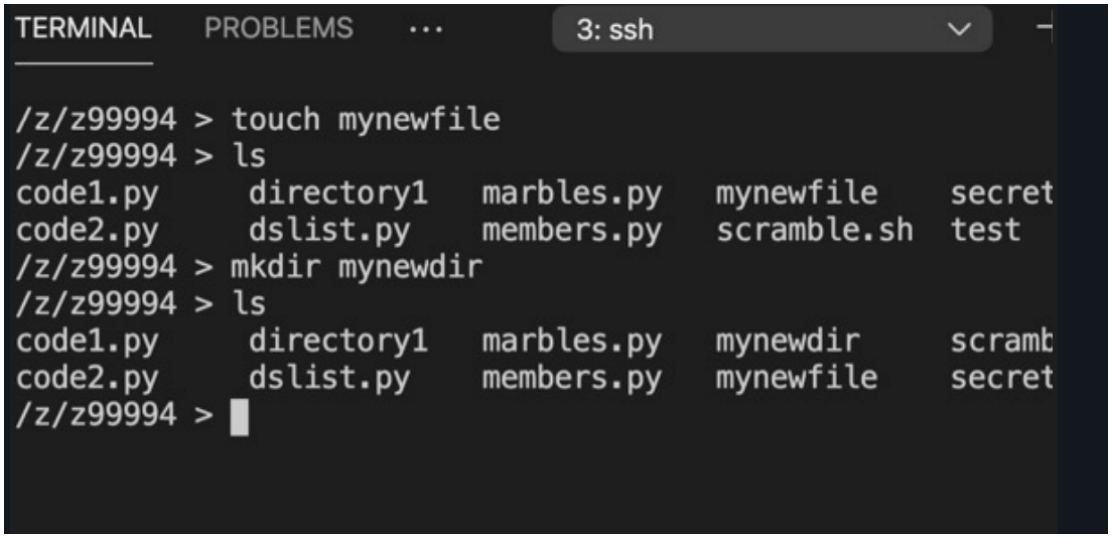
7 FRESH FILES AND FOLDERS

The `touch` command is commonly used to update the “last modified” timestamp of a file but can also be used to create an empty file.

Enter the command `touch mynewfile` and follow that up with a `ls` to see the brand new file.

You can create brand new directories with the `mkdir` command.

For example, try `mkdir mynewdir` and you should see a shiny new directory afterwards when you run `ls` again.

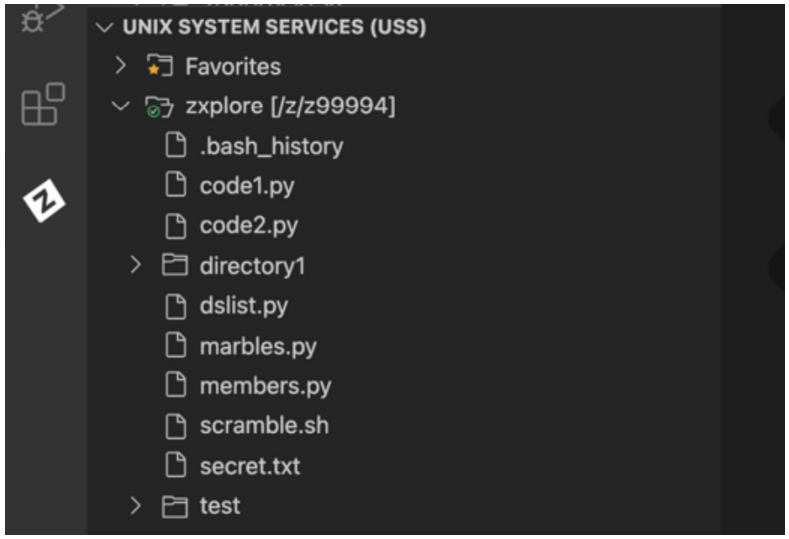


8 INVITE ZOWE TO THE PARTY

You can view your USS files and directories through Zowe.

Just click the Magnifying Glass next to your profile in the USS view and enter the full path of your home directory.

Your home directory is /z/zxxxxx



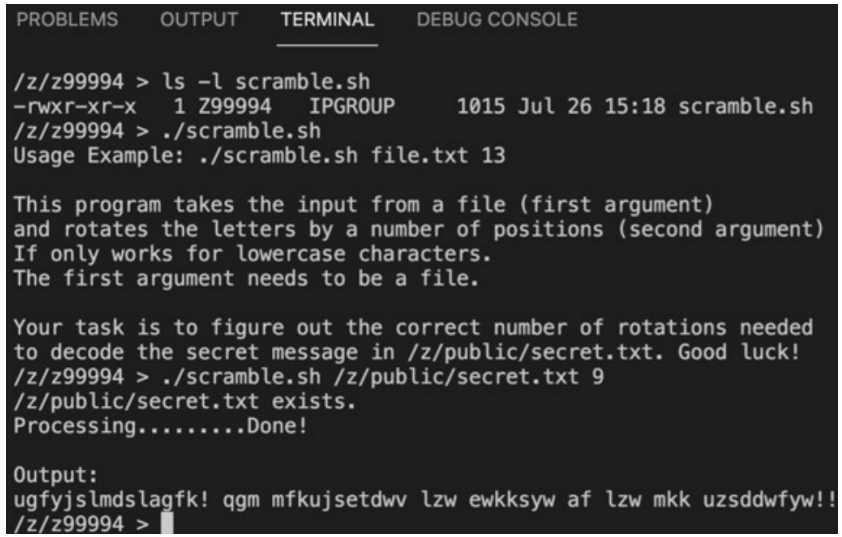
Make sure you use **all lowercase letters**, and that you use *your* userid. (Not zxxxxx or z99994)

9 I'VE GOT A SECRET

You have a program called `scramble.sh` in your home directory.

You can tell this is an executable program because when you enter the command `ls –l` it shows up with an 'x' in the fourth spot of the permissions. This means that in addition to you being able to *R*ead and *W*rite it, you can also e*X*ecute it.

There's a lot more to know about permission bits, but we'll save that for a more advanced USS lesson, perhaps.



For now, just know that you can run the program with the command

`./scramble.sh`

and the output of the program will tell you everything you need to know. Good luck!

Hint: The correct value for the number of rotations is somewhere between 1-26. Use your skills of deduction to try and figure out the value in as few tries as possible.

“I MADE A BUNCH OF FILES AND FOLDERS; HOW DO I GET RID OF THEM?”

To remove a file, type `rm` followed by the name of the file you want to delete. For example, `rm mynewfile`

You can also use the command `rmdir` to erase an empty directory. For example, `rmdir mynewdir` will get rid of that directory you created in Step #7.

There are ways to specify non-empty directories, as well as ways you can delete lots of files all at once. If you have some experience in a shell environment, you probably know these commands. If not, it’s probably best we don’t show them to you this early. We’d hate to see you delete all of your hard work.

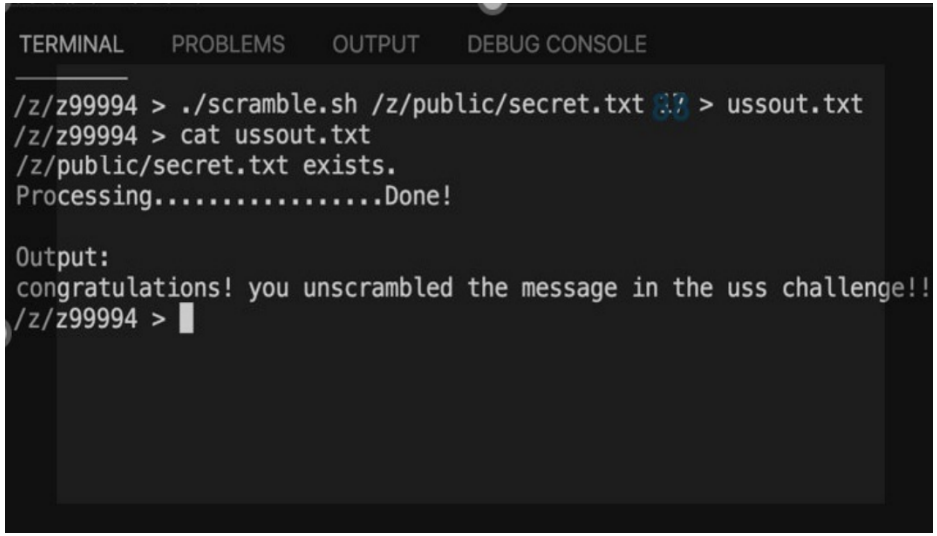
If you *do* accidentally delete a file, you can usually find the original copy of it in `/z/public`. Use the `cp` command to copy it.

Example: `cp /z/public/test ~/test` will give you a fresh copy of ‘test’ in your home directory.

USS11250417-2338

10 REDIRECTING THE OUTPUT

Now you have cracked the code, put the output of the program into a file. This is super easy to do, using “redirection”.



Type, or recall, your recent successful scramble command with the correct values, and add `> ussout.txt` to the end of the command, so it will look like the screenshot above.

You won’t see output while the command is working.

When you put ‘>’ after a command, or anything that produces output, it means “Instead of putting the output to the screen, save it in this file instead”.

If the file doesn’t already exist, a new one will be created for you, but be careful … *redirection will also overwrite anything in an existing file*.

You can verify the output with the `cat ussout.txt` command.

You can use **‘>>’** to redirect the output to append to the bottom/end of the output file, which is exactly what we’ll do in Step #11.

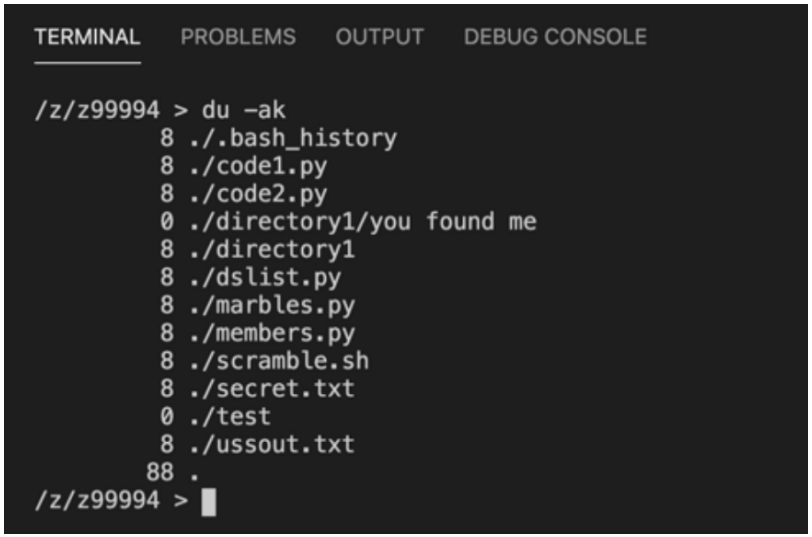
You can check on what’s in the file by opening it with VSCode in the USS view.

(Keep in mind you may need to Right-click and select “Pull from Mainframe” to refresh the view after writing out to it)

11 SPACE EXPLORATION

Once you have the decoded output in your `ussout.txt` file, use redirection to append to the end of that file (do *not* overwrite!) with the output of `du –ak`

The `du` command outputs the *D*isk *U*sage of the directory you are currently in, as well as all of the directories inside/below that directory (if you specify the '-a' option) and will give you the output in kilobytes (that is what the '-k' option is for).



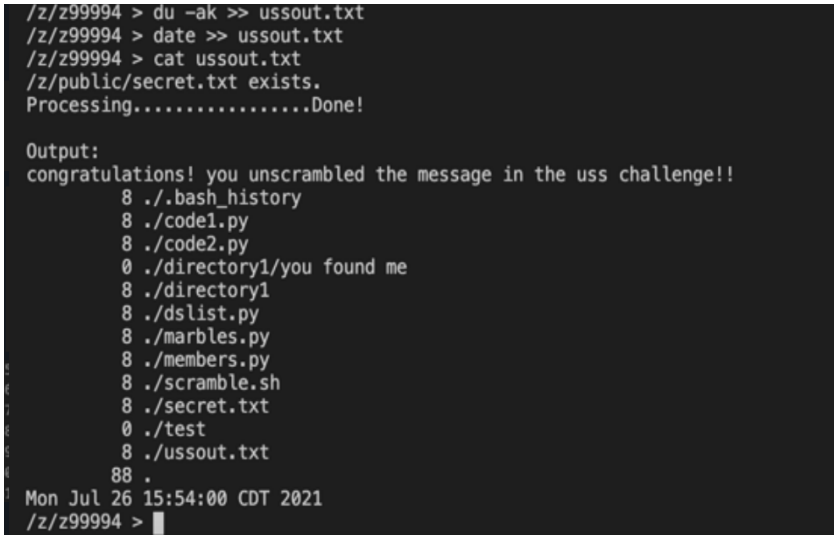
The output from this command will vary as more we provide more samples. (Don't worry if the screenshot is different for your results.)

If you have completed this correctly, your `ussout.txt` file should have:

1. secret decoded message, followed by
2. the output of your disk usage command.

Check the contents of your output with `cat ussout.txt`

12 MAKE IT COUNT



Finally, use redirection to append the output of the `date` command into the same file.

Make sure your **ussout.txt** file now has

1. the decoded secret message
2. your disk usage output
3. the output of the `date` command

If it all looks good, then submit the validation job **CHKUSS1** from **ZXP.PUBLIC.JCL**, and if all goes well, you're done!

Nice job - let's recap	Next up ...
You connected to USS through a secure shell (SSH) as well as through the Zowe plugin in VSCode. From there, you navigated directories, managed output, hacked some text, and learned all about arguments. It may not seem like much, but think of what you've done, and what you knew when you started. If you're into the UNIX way of doing things, you will probably really enjoy the LINUX-based challenge which is made available later on. If this wasn't your "cup of tea", at least now you're done!	We're going to keep working in the USS space for the next challenge, so make sure you keep those terminal commands and VSCode tabs handy. We'll be doing some very basic coding using Python. Don't panic if you've never coded before! We'll show you everything you need.