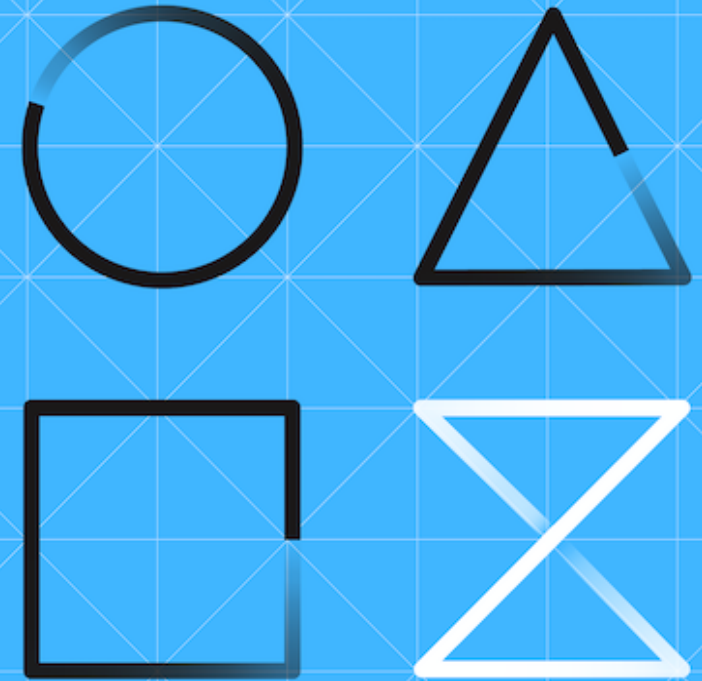


USS1

Let's Dive Deeper

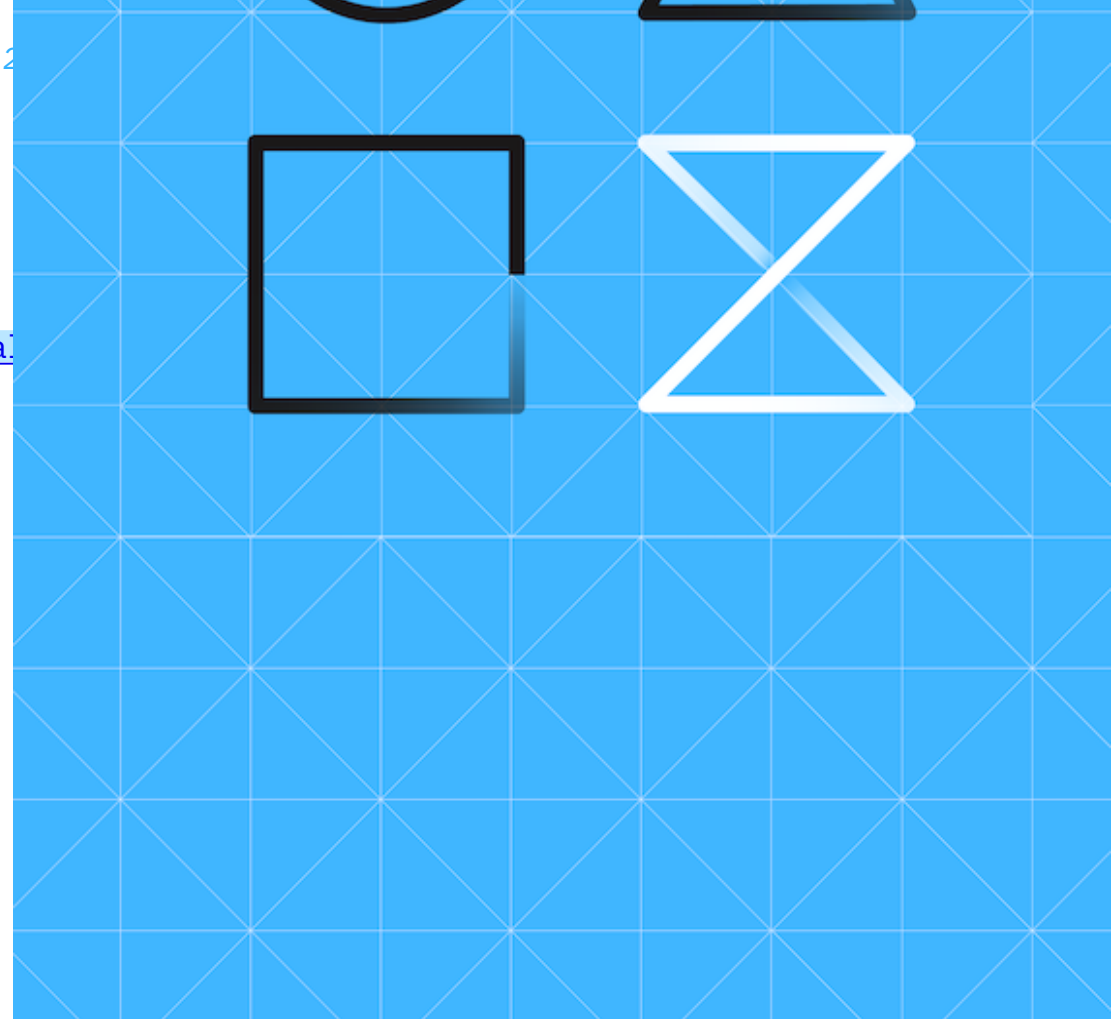
Automated Translation by Watson Language Translator



USS1

Profundicemos

- `cd circle & & cd .. & & cd-& & ./shake_it_a`
 - El reto
 - Antes de empezar
 - Inversiones
- 1 Buscar el terminal
- 2 Conectar a través de SSH
- 3 Verifíqueme, gracias
- 4 ¡Estás en! ¿Y ahora qué?
- 5 Orientación
- 6 CAMBIAR DIRECTORIO CON CD
- 7 Archivos y carpetas nuevos
- 8 INVITAR A ZOWE A LA FIESTA
- 9 Tengo un secreto
- 10 Redirección de la salida
- 11 Exploración de espacio
- 12 hacer que cuente



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

El Reto

UNIX System Services (USS) es un [POSIX](#) Implementación compatible con un entorno UNIX en z/OS que permite una experiencia similar a la de UNIX mientras se siguen utilizando las mismas API del sistema que el z/OS que ha estado utilizando hasta ahora.

Todo eso tendrá más sentido a medida que se abre paso a través de la shell, ejecutar conchas scripts, y caer irremediabilmente en el amor con la terminación de la pestaña.

Antes De Empezar

Estamos empezando con una faceta completamente nueva de z/OS, por lo que realmente no necesita saber mucho más, pero necesitará su entorno VSCode.

Un conocimiento de cómo funcionan los conjuntos de datos y los miembros sin duda ayudará a poner las cosas en perspectiva a medida que aprenda sobre USS.

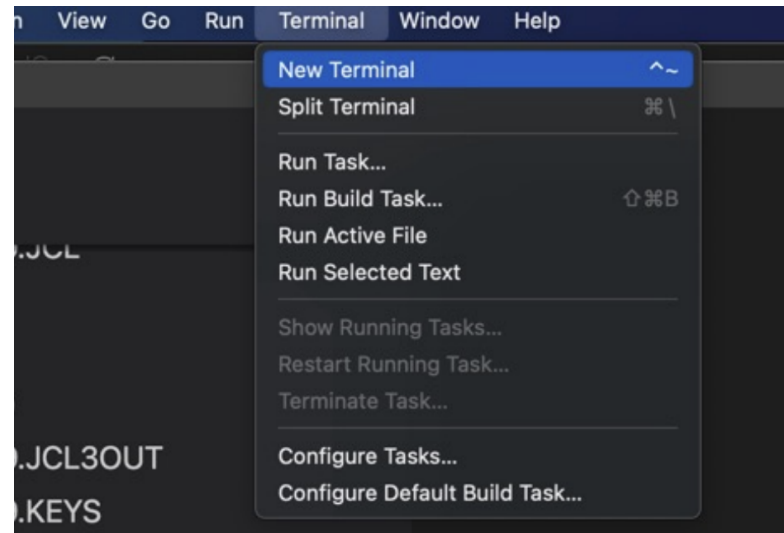
Inversiones

Pasos	Duración
12	60 minutos

1 BUSCAR EL TERMINAL

Busque la sección Terminal en la parte inferior de la ventana de VSCode.

Si no lo ve, intente utilizar la opción de menú Terminal y seleccione "Nuevo terminal".



Este es un método basado en texto para interactuar con su propio sistema personal.

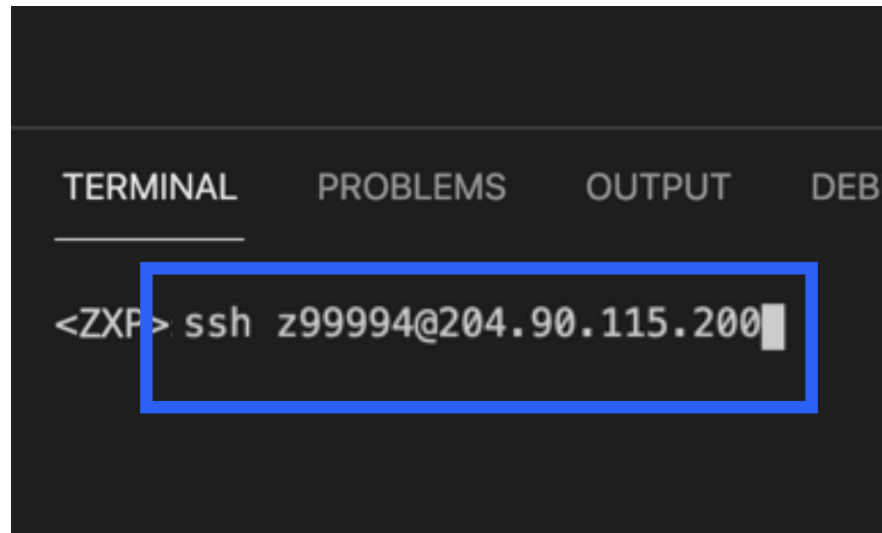
Si tiene una versión anterior de Windows, es posible que tenga que descargar un cliente SSH independiente, como por ejemplo [PuTTY](#) .

Para obtener orientación al respecto, consulte los foros de soporte.

2 CONECTAR A TRAVÉS DE SSH

Inicie sesión en el sistema z/OS con el mandato siguiente:

`ssh zxxxxx@204.90.115.200` (sustituyendo zxxxxx por su propio ID de usuario).



Para decirlo de otro modo, este mandato dice "Utilice el mandato ssh para conectarme (utilizando mi ID de usuario) al sistema remoto (en la dirección IP de destino)"

La primera vez que se conecte, puede ver un mensaje sobre la aprobación de la conexión continuada:

```
The authenticity of host '204.90.115.200 (204.90.115.200)' can't be established.  
RSA key fingerprint is SHA256:1YtEA18or6MI0VQnVQn7ZUCtFVkJMRStN+DnqJZaxPk.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? █
```

Esto es normal-responder "sí", y continuar.

3 VERIFÍQUEME, GRACIAS

Ahora se le solicitará la contraseña, que es la misma contraseña que ha utilizado para iniciar sesión en el sistema z/OS a través de VSCode.

Algunas cosas a tener en cuenta:

- Usted puede ver "Autenticidad de host ... no se puede establecer"
- Es posible que se le solicite que confíe o acepte una clave del sistema remoto.

Puede responder con seguridad "sí" a cualquiera de estas solicitudes.

No verá ningún carácter mientras escribe la contraseña.

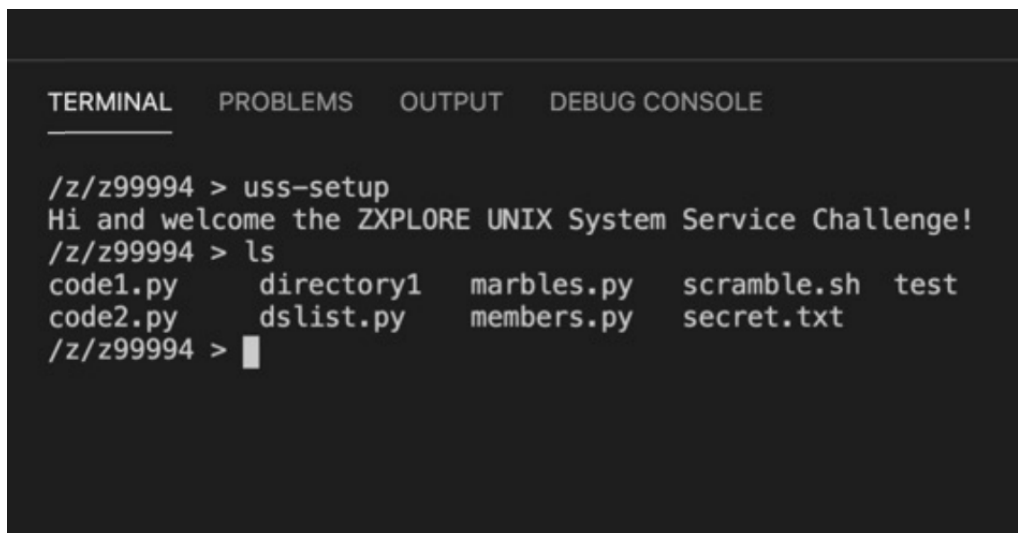
Esto es para que la gente que mira por encima de su hombro no pueda robar su contraseña, pero el sistema todavía puede verlo.

4 ¡ESTÁS EN! ¿Y AHORA QUÉ?

Ahora que ha iniciado sesión en el entorno USS con ssh, puede mirar a su alrededor con el mandato `ls`.

Esto mostrará todos los archivos y carpetas (también conocidos como directorios en UNIX-land).

Su valor predeterminado ("inicio") el directorio estará vacío-especifique el mandato `uss-setup` y, a continuación, ejecute el mandato `ls` de nuevo.



```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

/z/z99994 > uss-setup
Hi and welcome the ZXPLORÉ UNIX System Service Challenge!
/z/z99994 > ls
code1.py      directory1    marbles.py   scramble.sh   test
code2.py      dslist.py    members.py   secret.txt
/z/z99994 > 
```

USS utiliza una estructura jerárquica, donde hay archivos y directorios dentro de otros directorios.

Usted puede estar acostumbrado a este tipo de sistema de archivos en su propio ordenador, donde puede mantener los archivos en carpetas, y poner las carpetas dentro de las carpetas y así sucesivamente.

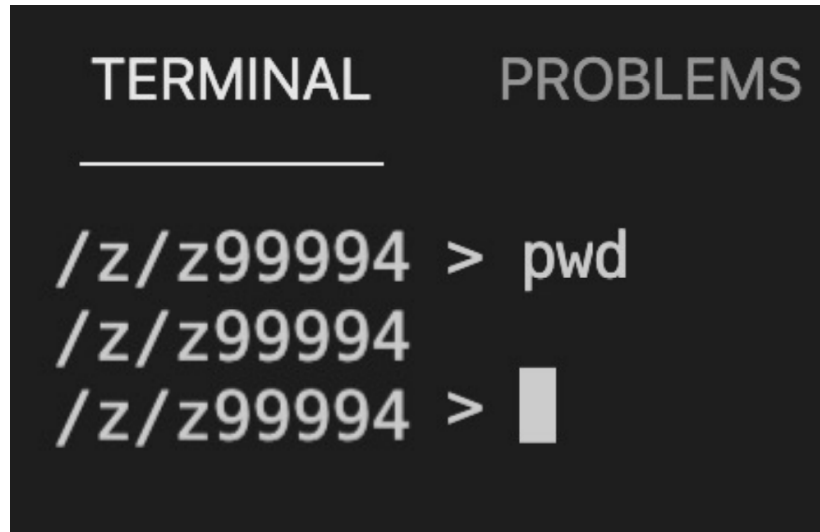
Cuando desee desconectarse, sólo tiene que especificar el mandato `exit` y se cerrará la sesión. Puede utilizar el mandato ssh desde su portátil para volver a conectarse más tarde.

Si deja la sesión ssh desocupada durante más de unos minutos (normalmente 3-5 minutos), no responderá. Si eso sucede, su terminal aparecerá atascado. Para borrar esto, escriba "~."-es posible que tenga que hacer esto un par de veces.

5 ORIENTACIÓN

Además de mirar a su alrededor con `ls`, usted también puede querer saber "¿Dónde estoy?".

El indicador de mandatos predeterminado normalmente le mostrará dónde se encuentra en la estructura de directorios, pero también puede escribir `pwd` a P pinta el E orking D irectorio.



```

  TERMINAL      PROBLEMS
  -----
/z/z99994 > pwd
/z/z99994
/z/z99994 > █

```

Esto será útil en breve cuando empiece a utilizar `cd` a C Hange D irectory para moverse por el sistema de archivos.

En este momento, está en su directorio de inicio, que es donde su Archivos USS activos.

Puede volver a este directorio en cualquier momento escribiendo `cd ~` (esa es la clave de tilde)

6 CAMBIAR DIRECTORIO CON CD

Para navegar a otro directorio, escriba `cd`, seguido del nombre del directorio.

Por ejemplo, podemos escribir `cd directory1` y vamos a entrar en `directory1`, suponiendo que es un directorio que podemos ver con el `ls` mandato.

Pruébalo y, a continuación, escriba `pwd`

Debería ver que `pwd` ahora muestra su ruta como `/z/zxxxxx/directory1`

Para volver a su directorio de inicio, necesita volver a un nivel. Puede hacerlo escribiendo `cd ..` (dos puntos), o puede utilizar el atajo de tilde para ir directamente a su directorio de inicio desde cualquier lugar (`cd ~`).

Hasta ahora, ustedes se han estado moviendo de un lado a otro, un paso a la vez. También puede cambiar de directorio especificando la vía de acceso completa al directorio al que desea ir.

Por ejemplo, `cd /z/public/test` le llevará directamente a esa nueva ubicación.

```
/z/z99994 > cd directory1/
/z/z99994/directory1 > pwd
/z/z99994/directory1
/z/z99994/directory1 > cd ..
/z/z99994 > pwd
/z/z99994
/z/z99994 > cd directory1/
/z/z99994/directory1 > cd ~
/z/z99994 > pwd
/z/z99994
/z/z99994 > cd /z/public/test/
/z/public/test > pwd
/z/public/test
/z/public/test > cd ~
/z/z99994 > pwd
/z/z99994
/z/z99994 >
```

A continuación, utilice `cd ~` para volver a casa, después de que hemos ido allí y miramos alrededor un poco.

(De nuevo, si el directorio de inicio está vacío, ejecute el mandato `uss-setup` mandato)

HACER LA VIDA MAS FACIL. UTILIZAR FINALIZACIÓN DE PESTAÑA Y FLECHAS ARRIBA/ABAJO

Observe cómo la palabra "directorio" se pone molesto para tener que escribir después de un tiempo? Pruebe este truco limpio ... la próxima vez que tenga que escribir un largo mandato o nombre de archivo, escriba las primeras letras, a continuación, pulse la tecla Tab.

Al golpearlo una vez se completará automáticamente la mayor parte del mandato o el nombre que pueda, y al golpearlo dos veces le mostrará todas las finalizaciones posibles. Por lo tanto, si tiene un "directory1" y un "directory2", puede escribir `cd di` seguido por pulsar la tecla de tabulación, y se auto-completar la parte "directorio".

Pulse una vez más y le dirá "hay directorio1 y directorio 2".

Déle una oportunidad, y se sorprenderá de lo mucho más rápido que puede llegar con el indicador de comandos.

Un truco más, si quieres usar un comando que tecleaste no hace mucho tiempo, puedes recordar comandos recientes pulsando la flecha Arriba en tu teclado. A continuación, simplemente pulse Intro para utilizarlo. Bastante dulce, ¿verdad?

Si necesita modificar un mandato anterior, utilice la tecla de retroceso para eliminar los caracteres no deseados y, a continuación, escriba lo que necesita.

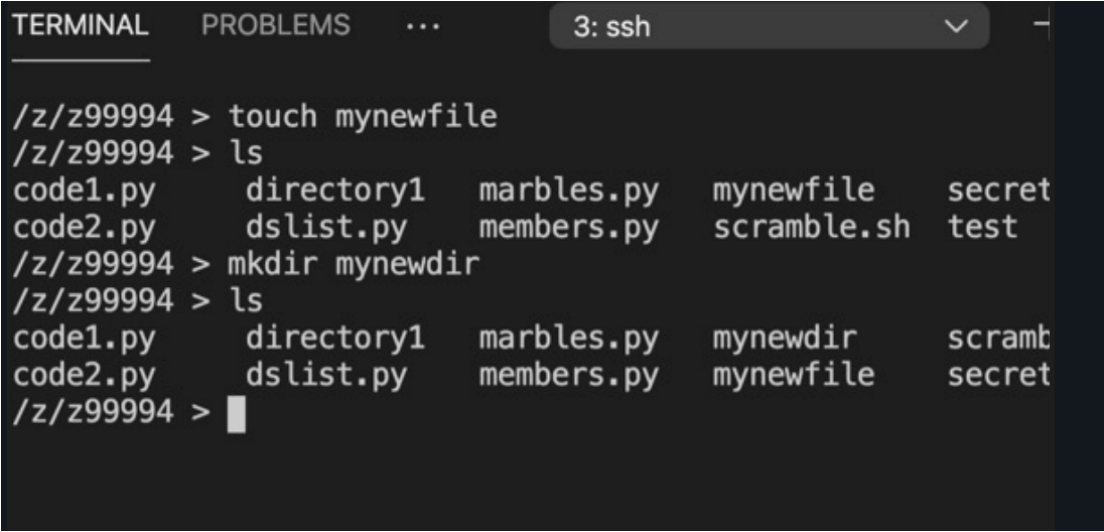
7 ARCHIVOS Y CARPETAS NUEVOS

El `touch` Normalmente se utiliza para actualizar la indicación de fecha y hora "última modificación" de un archivo, pero también se puede utilizar para crear un archivo vacío.

Especifique el mandato `touch mynewfile` y haga un seguimiento de ello con un `ls` para ver el nuevo archivo.

Puede crear nuevos directorios con el `mkdir` mandato.

Por ejemplo, intente `mkdir mynewdir` y debería ver un nuevo directorio brillante después cuando ejecute `ls` de nuevo.



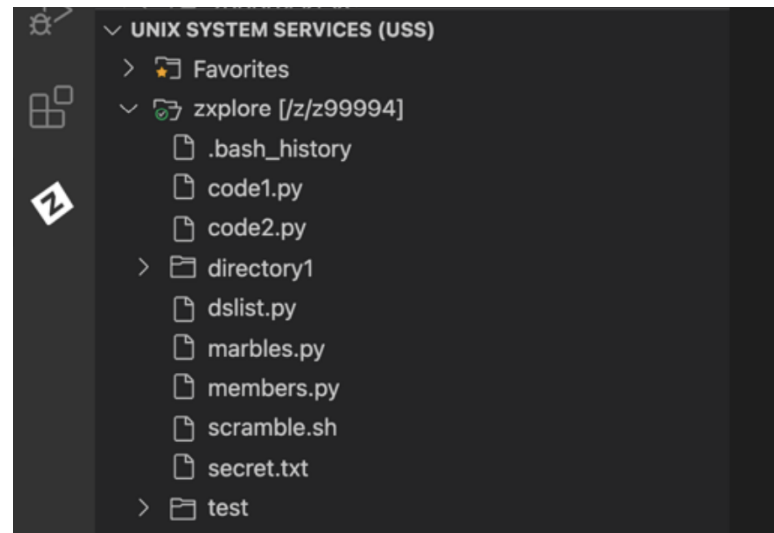
```
TERMINAL  PROBLEMS  ...  3: ssh  v  -
/z/z99994 > touch mynewfile
/z/z99994 > ls
code1.py    directory1  marbles.py  mynewfile   secret
code2.py    dslist.py  members.py  scramble.sh  test
/z/z99994 > mkdir mynewdir
/z/z99994 > ls
code1.py    directory1  marbles.py  mynewdir    scramb
code2.py    dslist.py  members.py  mynewfile   secret
/z/z99994 > █
```

8 INVITAR A ZOWE A LA FIESTA

Puede ver los archivos y directorios USS a través de Zowe.

Simplemente haga clic en la lupa junto a su perfil en la vista USS y especifique la ruta completa de su directorio de inicio.

El directorio de inicio es `/z/zxxxxx`



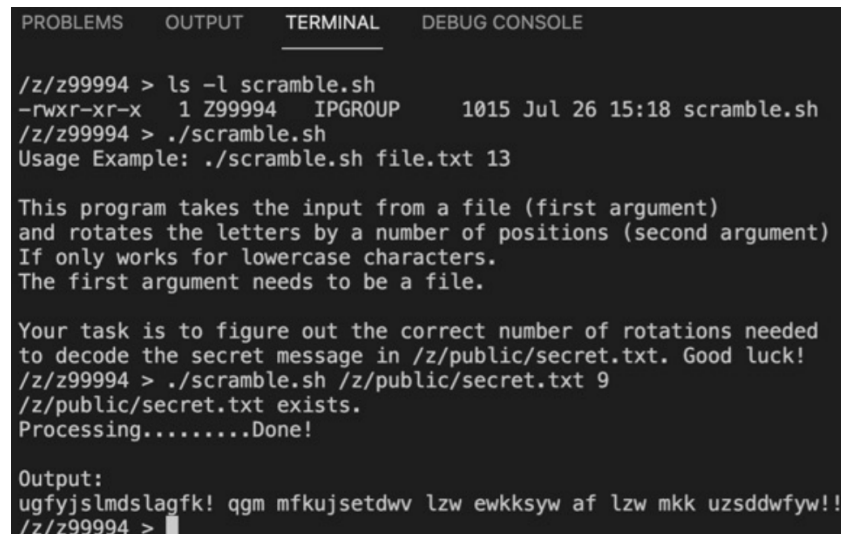
Asegúrese de utilizar **todas las letras minúsculas** , y que utilice su ID de usuario. (No zxxxxx ni z99994)

9 TENGO UN SECRETO

Tiene un programa llamado `scramble.sh` en el directorio de inicio.

Puede indicar que se trata de un programa ejecutable porque al especificar el mandato `ls -l` se muestra con una 'x' en el cuarto lugar de los permisos. Esto significa que además de ser capaz de *R*ead y *E*rite, también puede *X* Ecuéntate.

Hay mucho más que saber sobre los bits de permiso, pero guardaremos eso para una lección más avanzada de USS, tal vez.



The screenshot shows a terminal window with the following content:

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

/z/z99994 > ls -l scramble.sh
-rwxr-xr-x  1 Z99994  IPGROUP      1015 Jul 26 15:18 scramble.sh
/z/z99994 > ./scramble.sh
Usage Example: ./scramble.sh file.txt 13

This program takes the input from a file (first argument)
and rotates the letters by a number of positions (second argument)
If only works for lowercase characters.
The first argument needs to be a file.

Your task is to figure out the correct number of rotations needed
to decode the secret message in /z/public/secret.txt. Good luck!
/z/z99994 > ./scramble.sh /z/public/secret.txt 9
/z/public/secret.txt exists.
Processing.....Done!

Output:
ugfyjslmdslagfk! qgm mfkujsedwv lzw ewkksyw af lzw mkk usdddwfyw!!
/z/z99994 >
```

Por ahora, sólo tiene que saber que puede ejecutar el programa con el mandato

`./scramble.sh`

y la salida del programa le dirá todo lo que necesita saber. ¡Buena suerte!

Sugerencia : El valor correcto para el número de rotaciones está entre 1-26. Utilice sus habilidades de deducción para tratar de averiguar el valor en tan pocos intentos como sea posible.

"HICE UN MONTÓN DE ARCHIVOS Y CARPETAS; ¿CÓMO ME DESHAGO DE ELLOS?"

Para eliminar un archivo, escriba `rm` seguido del nombre del archivo que desea suprimir. Por ejemplo, `rm mynewfile`

También puede utilizar el mandato `rmdir` para borrar un directorio vacío. Por ejemplo, `rmdir mynewdir` se deshacerá del directorio que ha creado en el paso #7.

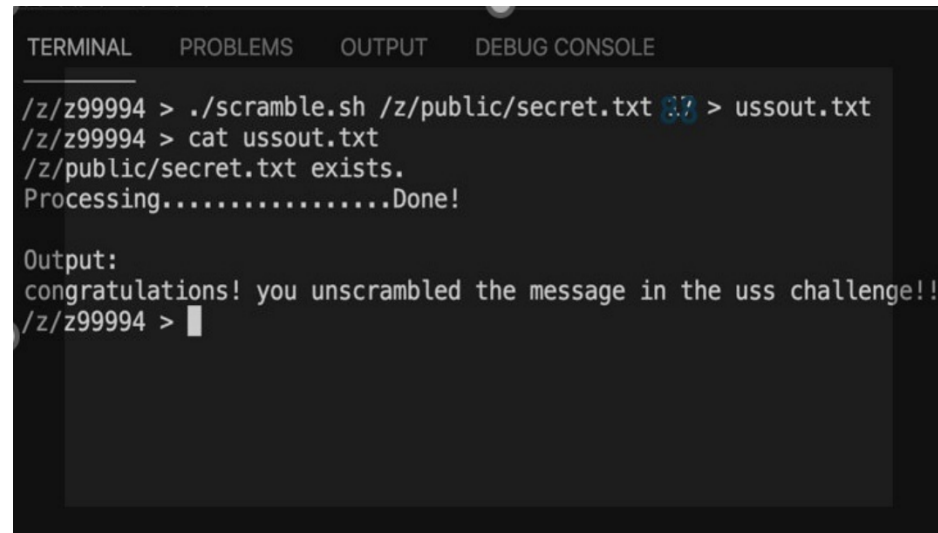
Hay formas de especificar directorios no vacíos, así como formas de eliminar muchos archivos de una sola vez. Si tiene alguna experiencia en un entorno de shell, probablemente conozca estos mandatos. Si no es así, probablemente es mejor que no se los mostremos tan pronto. Nos gustaría ver que borra todo su trabajo duro.

Si *hacer* suprimir accidentalmente un archivo, normalmente puede encontrar la copia original del mismo en `/z/public`. Utilice el `cp` para copiarlo.

Ejemplo : `cp /z/public/test ~/test` le dará una copia nueva de 'test' en su directorio de inicio.

10 REDIRECCIÓN DE LA SALIDA

Ahora que ha roto el código, ponga la salida del programa en un archivo. Esto es muy fácil de hacer, usando "redirección".



```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

/z/z99994 > ./scramble.sh /z/public/secret.txt > ussout.txt
/z/z99994 > cat ussout.txt
/z/public/secret.txt exists.
Processing.....Done!

Output:
congratulations! you unscrambled the message in the uss challenge!!
/z/z99994 > 
```

Escriba, o recupere, el mandato scramble satisfactorio reciente con los valores correctos y añada `> ussout.txt` al final del comando, por lo que se parecerá a la captura de pantalla anterior.

No verá la salida mientras el mandato esté funcionando.

Cuando ponga `'>'` después de un comando, o cualquier cosa que produzca salida, significa "en lugar de poner la salida en la pantalla, guardarla en este archivo en su lugar".

Si el archivo todavía no existe, se creará uno nuevo para usted, pero tenga cuidado ... *redirección también sobrescribirá cualquier cosa en un archivo existente* .

Puede verificar la salida con el `cat ussout.txt` mandato.

Puede utilizar "> >" para redirigir la salida para añadirla a la parte inferior/final del archivo de salida, que es exactamente lo que haremos en el paso #11.

Puede comprobar qué hay en el archivo abriéndolo con VSCode en la vista USS.

(Tenga en cuenta que es posible que tenga que pulsar con el botón derecho del ratón y seleccionar "Extraer del sistema principal" para renovar la vista después de escribir en él)

11 EXPLORACIÓN DE ESPACIO

Una vez que tenga la salida decodificada en su `ussout.txt` archivo, utilice la redirección para añadir al final de ese archivo (do *no* sobrescribir!) con la salida de `du -ak`

El `du` salida del mandato *D isk I sage* del directorio en el que se encuentra actualmente, así como todos los directorios dentro/debajo de ese directorio (si especifica la opción '-a') y le dará la salida en kilobytes (para eso es la opción '-k').

```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

/z/z99994 > du -ak
8  ./bash_history
8  ./code1.py
8  ./code2.py
0  ./directory1/you found me
8  ./directory1
8  ./dslist.py
8  ./marbles.py
8  ./members.py
8  ./scramble.sh
8  ./secret.txt
0  ./test
8  ./ussout.txt
88 .
/z/z99994 > █
```

En este punto, debe utilizar menos de 50 kilobytes en total en la estructura de directorios de inicio.

Si lo ha completado correctamente, el archivo `ussout.txt` debería tener:

1. mensaje decodificado secreto, seguido de
2. la salida del mandato de uso de disco.

Compruebe el contenido de la salida con `cat ussout.txt`

12 HACER QUE CUENTE

```
/z/z99994 > du -ak >> ussout.txt
/z/z99994 > date >> ussout.txt
/z/z99994 > cat ussout.txt
/z/public/secret.txt exists.
Processing.....Done!

Output:
congratulations! you unscrambled the message in the uss challenge!!
 8 ./bash_history
 8 ./code1.py
 8 ./code2.py
 0 ./directory1/you found me
 8 ./directory1
 8 ./dslist.py
 8 ./marbles.py
 8 ./members.py
 8 ./scramble.sh
 8 ./secret.txt
 0 ./test
 8 ./ussout.txt
 88 .
Mon Jul 26 15:54:00 CDT 2021
/z/z99994 >
```

Por último, utilice la redirección para añadir la salida del `date` en el mismo archivo.

Asegúrese de que **ussout.txt** el archivo ahora tiene

1. el mensaje secreto decodificado
2. la salida de uso de disco
3. la salida de la date mandato

Si todo parece correcto, envíe el trabajo de validación **CHKUSS1** desde **ZXP.PUBLIC.JCL** , y si todo va bien, ¡ya está!

Buen trabajo-recapitulemos	Siguiente ...
<p>Se ha conectado a USS a través de un shell seguro (SSH), así como a través del plugin Zowe en VS Code. A partir de ahí, navegó por directorios, gestionó la salida, hackeó algún texto y aprendió todo sobre los argumentos. Puede que no parezca mucho, pero piensa en lo que has hecho, y en lo que sabías cuando empezaste. Si estás en la forma UNIX de hacer las cosas, probablemente disfrutarás realmente del reto basado en LINUX que está disponible más adelante. Si esta no era tu taza de té, ¡al menos ahora ya has terminado!</p>	<p>Vamos a seguir trabajando en el espacio USS para el siguiente desafío, así que asegúrate de mantener a mano esos comandos de terminal y las pestañas de VSCode. Estaremos haciendo una codificación muy básica usando Python. ¡No te asustes si nunca antes has codificado! Te mostraremos todo lo que necesitas.</p>