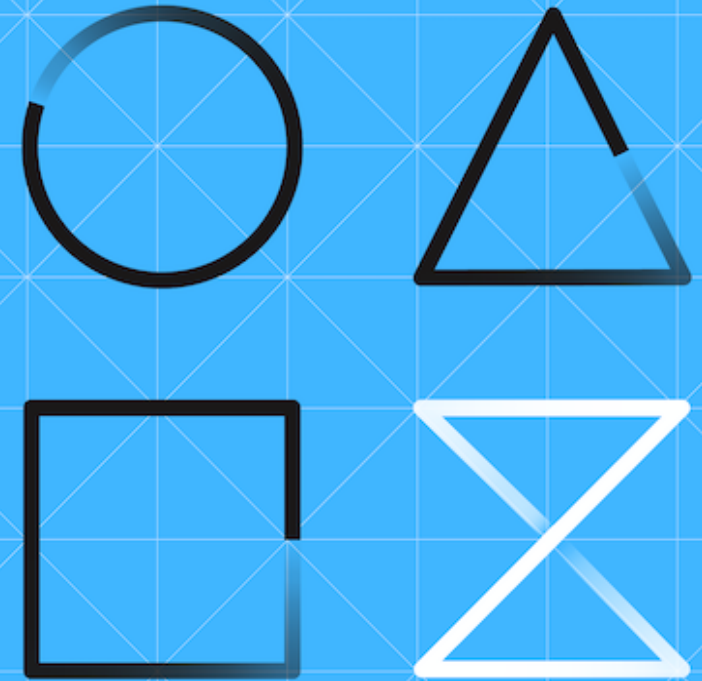


# USS1

Let's Dive Deeper

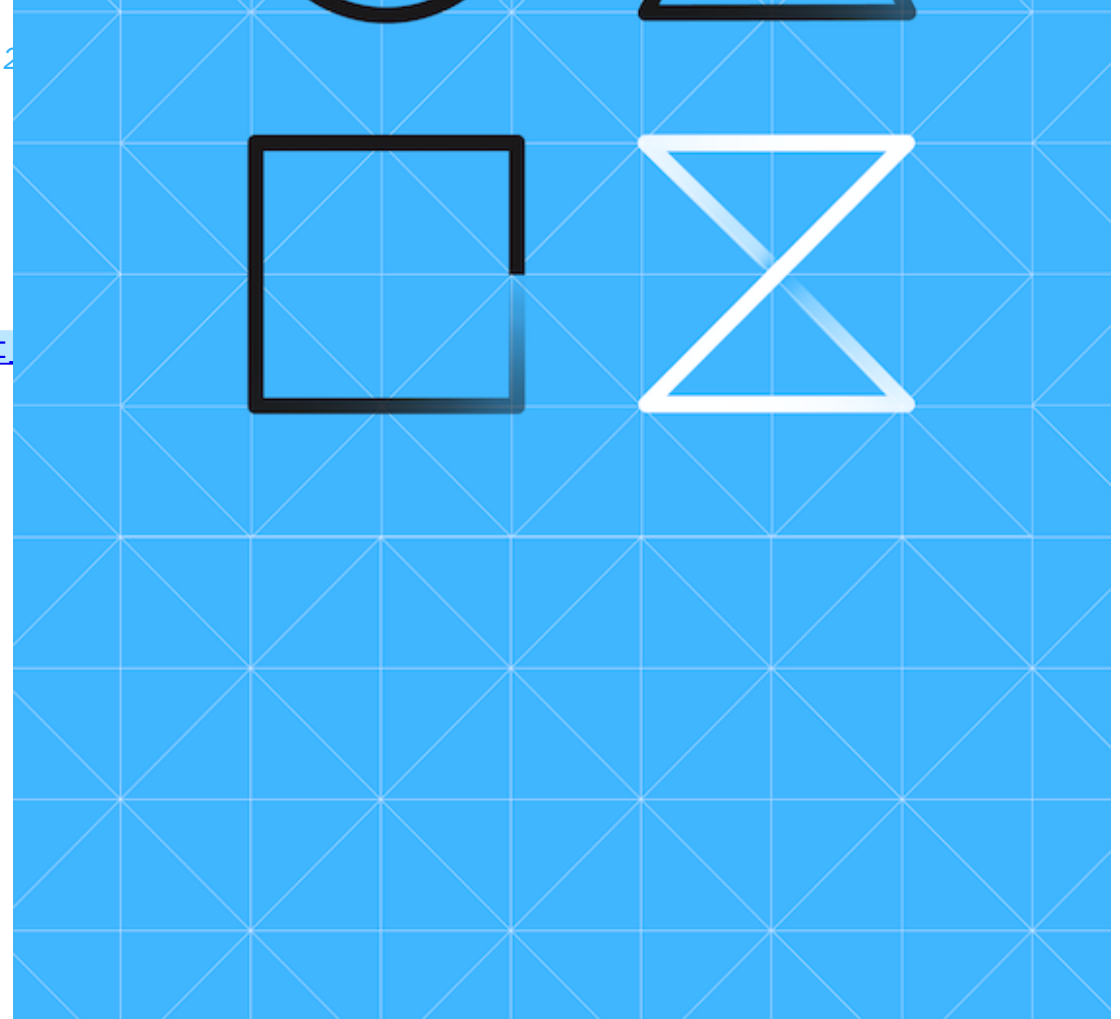
Automated Translation by Watson Language Translator



# USS1

## Vamos Dive Deeper

- [cd circle & & cd .. & & cd-& & ./shake\\_it](#)
  - [0 desafio](#)
  - [Antes de começar](#)
  - [Investimento](#)
- [1 Encontre o terminal](#)
- [2 Conectar por meio de SSH](#)
- [3 Verifique-me, obrigado.](#)
- [4 Você está dentro! E agora?](#)
- [5 Orientando-se](#)
- [6 ALTERAR DIRETÓRIO COM CD](#)
- [7 Arquivos e pastas atualizados](#)
- [8 CONVIDE ZOWE PARA A FESTA](#)
- [9 Eu tenho um segredo](#)
- [10 Redirecionando a saída](#)
- [11 Exploração espacial](#)
- [12 faça valer a pena](#)



# CD CIRCLE & & & CD .. & & CD-& & ./ SHAKE\_IT\_ALL\_ABOUT.SH

## 0 Desafio

UNIX System Services (USS) é um **POSIX** -compliant implementação de um ambiente UNIX no z/OS que permite uma experiência semelhante ao UNIX enquanto ainda usa as mesmas APIs do sistema que o z/OS que você tem usado até agora.

Tudo isso fará mais sentido à medida que você fizer o seu caminho através do shell, executar scripts de shells e se apaixonar irremediavelmente pela conclusão da aba.

## Antes De Começar

Estamos começando com uma nova faceta do z/OS, portanto, você não precisa saber muito mais, mas precisará do ambiente do VSCode.

Um conhecimento de como conjuntos de dados e membros trabalham certamente ajudará a colocar as coisas em perspectiva à medida que você aprende sobre o USS.

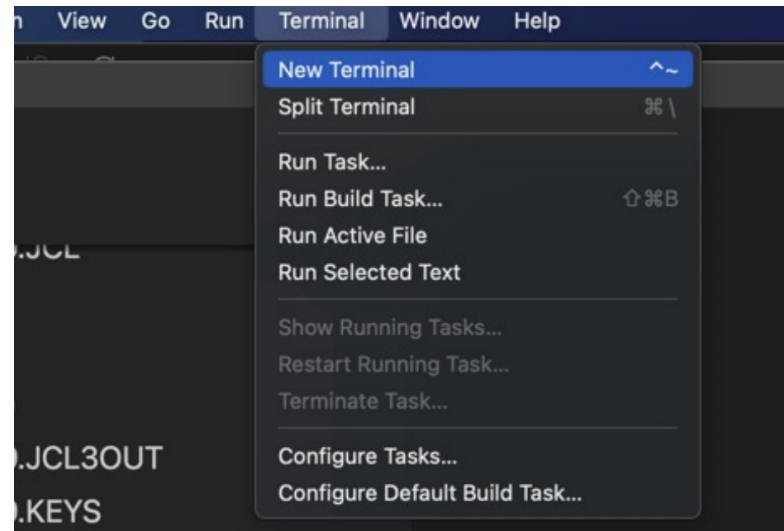
## Investimento

Etapas	Duração
12	60 minutos

# 1 ENCONTRE O TERMINAL

Procure a seção Terminal na parte inferior de sua janela do VSCode.

Se você não o vir, tente usar a opção de menu Terminal e selecionar "Novo Terminal".



Este é um método baseado em texto para interagir com o seu próprio sistema pessoal.

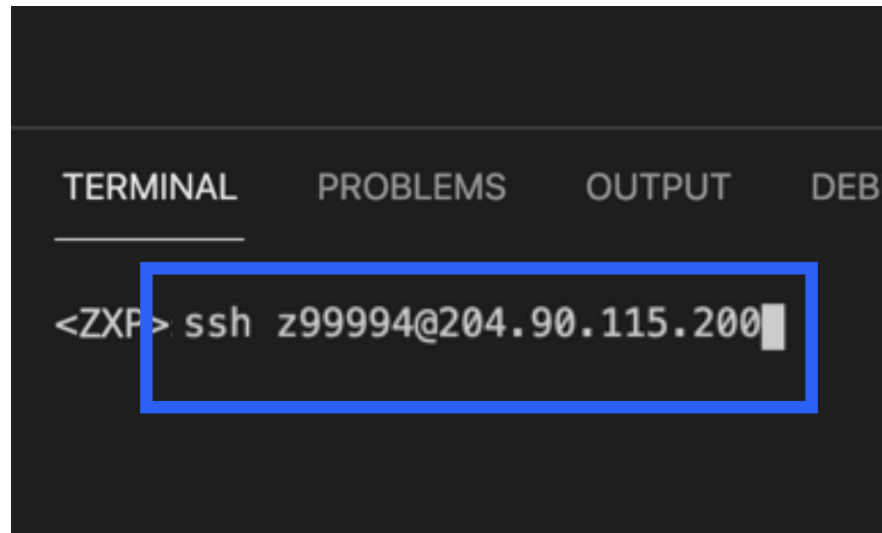
Se você tiver uma versão mais antiga do Windows, pode ser necessário fazer download de um cliente SSH separado, como [PuTTY](#).

Para orientação sobre isso, consulte os fóruns de suporte.

## 2 CONECTAR POR MEIO DE SSH

Efetue login no sistema z/OS com o comando a seguir:

`ssh zxxxxx@204.90.115.200` (substituindo o zxxxxx por seu próprio ID do usuário)



Para colocar outra maneira, este comando diz "Use o comando ssh para me conectar (usando meu ID do usuário) ao sistema remoto (no endereço IP de destino)"

Na primeira vez que você se conectar, você poderá ver uma mensagem sobre a aprovação da conexão 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])? █
```

Isso é normal-responda "sim" e continue.

### 3 VERIFIQUE-ME, OBRIGADO.

Agora será solicitada sua senha, que é a mesma senha que você usou para efetuar login no sistema z/OS por meio do VSCode.

Algumas coisas a serem observadas:

- Você pode ver "Autenticidade do host ... não pode ser estabelecida"
- Você pode ser solicitado a confiar ou aceitar uma chave do sistema remoto.

Você pode responder com segurança "sim" a qualquer um desses avisos.

**Você não verá nenhum caractere enquanto digita sua senha!**

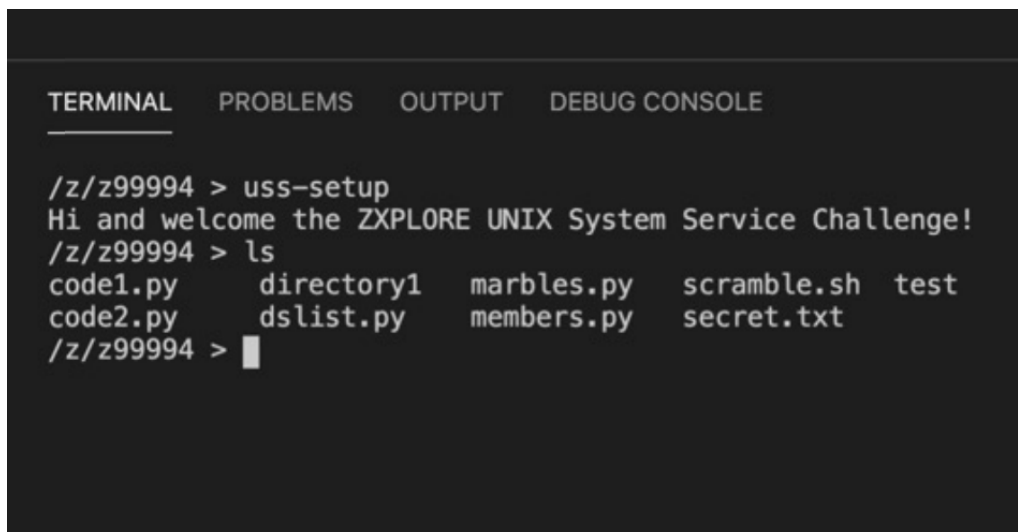
Isso é para que as pessoas olhando por cima do seu ombro não possam roubar sua senha, mas o sistema ainda pode vê-la.

## 4 VOCÊ ESTÁ DENTRO! E AGORA?

Agora que você efetuou login no ambiente USS com ssh, é possível dar uma olhada ao redor com o comando `ls`.

Isso mostrará todos os arquivos e pastas (também conhecidos como diretórios em UNIX-land).

Seu padrão ("home") o diretório estará vazio-insira o comando `uss-setup`, em seguida, execute o comando `ls` novamente.



```

  TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

/z/z99994 > uss-setup
Hi and welcome the ZXPLORE 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 usa uma estrutura hierárquica, onde há arquivos e diretórios dentro de outros diretórios.

Você pode ser usado para este tipo de sistema de arquivos em seu próprio computador, onde você pode manter arquivos em pastas, e colocar pastas dentro de pastas e assim por diante.

Quando você quiser desconectar, basta digitar o comando `exit` e você será desconectado. Você pode usar o comando ssh do seu laptop para se reconectar mais tarde.

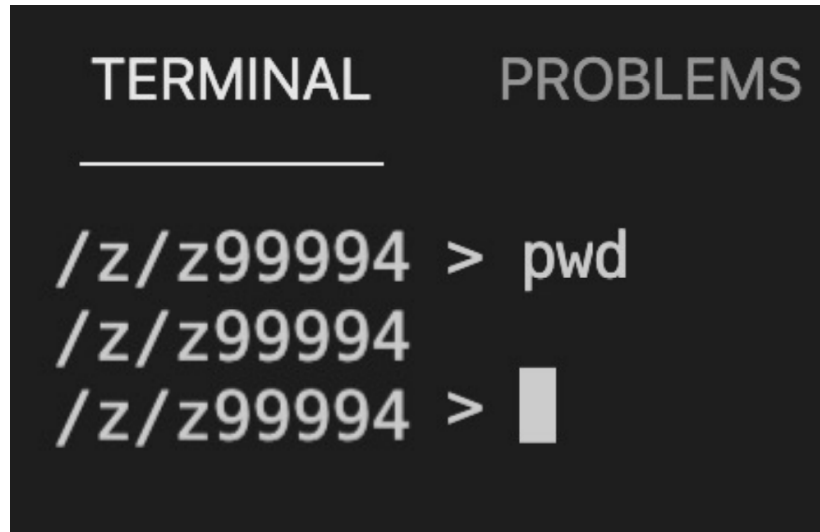


Se você deixar a sessão ssh inativa por mais de alguns minutos (geralmente 3-5 minutos), ela se tornará não responsiva. Se isso acontecer, seu terminal aparecerá preso. Para limpar isso, digite "~."-você pode precisar fazer isso algumas vezes.

## 5 ORIENTANDO-SE

Além de olhar em volta com `ls`, você também pode querer saber "Onde estou?".

O prompt de comandos padrão geralmente mostrará onde você está na estrutura de diretório, mas você também pode digitar `pwd` para *P*rint o *W*orking *D*ireitória.



A terminal window with a dark background. At the top, there are two tabs: 'TERMINAL' and 'PROBLEMS'. The 'TERMINAL' tab is active and has a horizontal line underneath it. Below the tabs, the terminal shows three lines of text: the first line is '/z/z99994 > pwd', the second line is '/z/z99994', and the third line is '/z/z99994 >' followed by a gray cursor bar.

Isso será útil em breve quando você começar a usar `cd` para *C*hange *D*iretory para mover em torno do sistema de arquivos.

Neste momento, você está em seu diretório inicial, que é onde o seu Arquivos USS ao vivo.

Você pode voltar a este diretório a qualquer momento digitando `cd ~` (essa é a chave de til)

## 6 ALTERAR DIRETÓRIO COM CD

Para navegar em outro diretório, digite `cd`, seguido pelo nome do diretório..

Por exemplo, podemos digitar `cd directory1` e vamos para o `directory1`, assumindo que é um diretório que podemos ver com o `ls` comando.

Experimente, em seguida, digite `pwd`

Você deve ver que o `pwd` agora mostra seu caminho como `/z/zxxxxxx/directory1`

Para voltar ao seu diretório inicial, você precisa voltar um nível. Você pode fazer isso digitando `cd ..` (dois pontos), ou você pode usar o atalho de til para ir direto para o seu diretório inicial a partir de qualquer lugar ( `cd ~` )

Até agora, você tem se movido para frente e para trás, um passo por vez. Também é possível mudar os diretórios especificando o caminho completo para o diretório para o qual você deseja ir

Por exemplo, `cd /z/public/test` levará você diretamente a esse novo local.

```
/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 >
```

Em seguida, use `cd ~` para voltar para casa, depois que nós fomos lá e olhamos em volta um pouco.

(Novamente, se seu diretório inicial estiver vazio, execute o `uss-setup` comando)

## TORNE A VIDA MAIS FÁCIL. USE A CONCLUSÃO DA GUIA E AS SETAS PARA CIMA / PARA BAIXO

Observe como a palavra "diretório" fica irritante para ter que digitar depois de um tempo? Tente este truque arrumado ... na próxima vez que você tiver que digitar um comando ou nome de arquivo longo, digite as primeiras letras e, em seguida, pressione a tecla Tab.

Atingi-lo uma vez irá auto-completar tanto do comando ou nome como ele pode, e bater duas vezes irá mostrar-lhe todas as conclusões possíveis. Então, se você tem um "directory1" e um "directory2", você pode digitar `cd di` seguido pressionando a tecla Tab, e ele concluirá automaticamente a parte do "diretório".

Pressione-o mais uma vez e ele irá dizer-lhe "há directory1 e directório 2".

Experimente, e você vai se surpreender com o quão mais rápido você pode obter com o prompt de comando.

Mais um truque, se você quiser usar um comando que você digitou não muito tempo atrás, você pode lembrar comandos recentes, pressionando a seta para cima em seu teclado. Em seguida, pressione Enter para usá-lo. Muito doce, certo?

Se você precisar alterar um comando anterior, use a tecla Backspace para remover caracteres indesejados e, em seguida, digite o que você precisa.

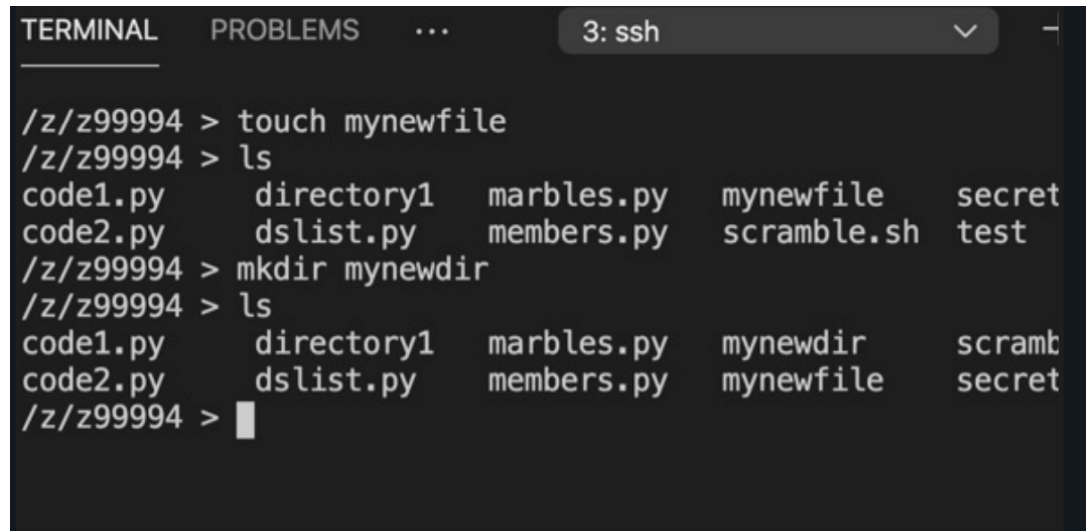
## 7 ARQUIVOS E PASTAS ATUALIZADOS

O `touch` é comumente usado para atualizar o registro de data e hora "última modificação" de um arquivo, mas também pode ser usado para criar um arquivo vazio.

Insira o comando `touch mynewfile` e seguir isso com um `ls` para ver o novo arquivo.

É possível criar novos diretórios com o `mkdir` comando.

Por exemplo, tente `mkdir mynewdir` e você deve ver um novo diretório brilhante depois quando você correr `ls` de novo.



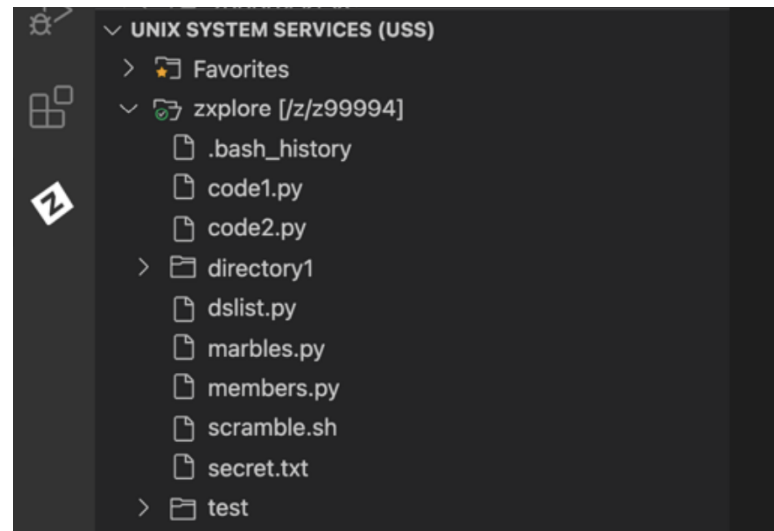
```
TERMINAL  PROBLEMS  ...  3: ssh  v  -
/z/z99994 > touch mynewfile
/z/z99994 > ls
code1.py    directory1  marbles.py  mynewfile   secret
code2.py    dslist.py  members.py  scramble.sh  test
/z/z99994 > mkdir mynewdir
/z/z99994 > ls
code1.py    directory1  marbles.py  mynewdir    scramb
code2.py  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

## 8 CONVIDE ZOWE PARA A FESTA

É possível visualizar seus arquivos e diretórios do USS por meio do Zowe

Basta clicar na Lupa ao lado do seu perfil na visualização USS e inserir o caminho completo do seu diretório inicial.

Seu diretório inicial é `/z/zxxxxx`



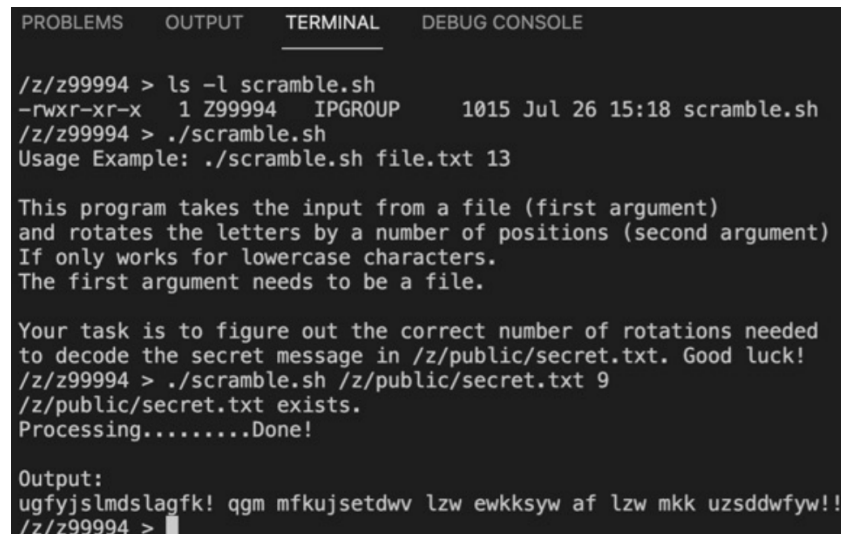
Certifique-se de usar **todas as letras minúsculas** , e que você usa o seu ID do usuário (Não zxxxxx ou z99994)

## 9 EU TENHO UM SEGREDO

Você tem um programa chamado `scramble.sh` no diretório inicial.

É possível dizer que este é um programa executável porque quando você insere o comando `ls -l` ele aparece com um 'x' no quarto lugar das permissões. Isso significa que, além de você ser capaz de *R*ead e *W*rite ele, você também pode *E*xecute o que é?

Há muito mais para saber sobre bits de permissão, mas vamos guardar isso para uma aula mais avançada do USS, talvez.



```
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 usddwfyw!!
/z/z99994 >
```

Por enquanto, apenas saiba que você pode executar o programa com o comando

`./scramble.sh`

e a saída do programa lhe dirá tudo o que você precisa saber. Boa sorte!

*Sugestão* : O valor correto para o número de rotações está entre 1 e 26. Use suas habilidades de dedução para tentar descobrir o valor em tão poucas tentativas quanto possível.



"Eu fiz um monte de arquivos e pastas; como eu me livrar deles?"

Para remover um arquivo, digite `rm` seguido pelo nome do arquivo que você deseja excluir. Por exemplo, `rm mynewfile`

Também é possível usar o comando `rmdir` para apagar um diretório vazio.. Por exemplo, `rmdir mynewdir` se livrará desse diretório criado na Etapa #7.

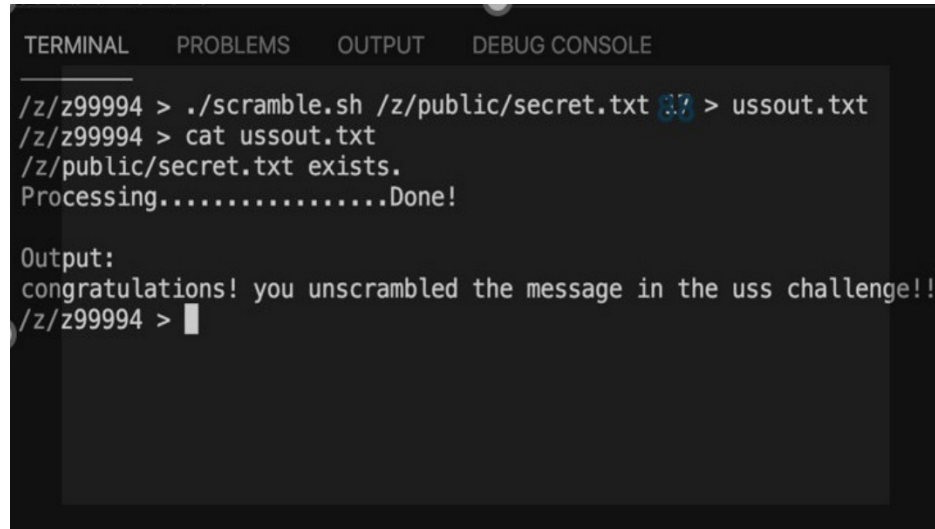
Há maneiras de especificar diretórios não vazios, bem como maneiras de excluir muitos arquivos de uma só vez. Se você tiver alguma experiência em um ambiente shell, provavelmente conhece esses comandos. Se não, provavelmente é melhor não mostrá-los a você tão cedo. Odiaríamos ver você apagar todo o seu trabalho duro.

Se você *faça* excluir acidentalmente um arquivo, você geralmente pode encontrar a cópia original dele em `/z/public`. Use o `cp` para copiá-lo.

*Exemplo :* `cp /z/public/test ~/test` dará uma cópia nova de 'test ' em seu diretório inicial.

## 10 REDIRECIONANDO A SAÍDA

Agora que você decifrou o código, coloque a saída do programa em um arquivo. Isto é super fácil de fazer, usando "redirecionamento".

A terminal window with tabs for 'TERMINAL', 'PROBLEMS', 'OUTPUT', and 'DEBUG CONSOLE'. The terminal shows a user at '/z/z99994' running './scramble.sh /z/public/secret.txt > ussout.txt'. The next command is 'cat ussout.txt', which outputs the message: '/z/public/secret.txt exists. Processing.....Done! Output: congratulations! you unscrambled the message in the uss challenge!!'. The prompt '/z/z99994 >' is shown at the bottom.

```
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 > 
```

Digite, ou rechame, seu comando de embaralhamento bem-sucedido recente com os valores corretos e inclua `> ussout.txt` para o fim do comando, para que ele se pareça com a captura de tela acima.

Você não verá saída enquanto o comando estiver funcionando.

Quando você coloca '>' após um comando, ou qualquer coisa que produza saída, significa "Em vez de colocar a saída na tela, salve-a nesse arquivo em vez disso".

Se o arquivo ainda não existir, um novo será criado para você, mas tenha cuidado ... *o redirecionamento também sobrescreverá qualquer coisa em um arquivo existente* .

É possível verificar a saída com o `cat ussout.txt` comando.

Você pode usar '>>' para redirecionar a saída para anexar à parte inferior / final do arquivo de saída, que é exatamente o que vamos fazer na Etapa #11

É possível verificar o que está no arquivo abrindo-o com VSCode na visualização USS.

(Tenha em mente que você pode precisar clicar com o botão direito do mouse e selecionar "Puxar do Mainframe" para atualizar a visualização depois de gravá-la)

# 11 EXPLORAÇÃO ESPACIAL

Depois de ter a saída decodificada em seu `ussout.txt`, use o redirecionamento para anexar ao final desse arquivo (do *não* sobrescrever!) com a saída de `du -ak`

O `du` saídas do comando *D* risco *U* sage do diretório em que você está atualmente, bem como todos os diretórios dentro / abaixo desse diretório (se você especificar a opção '-a') e lhe dará a saída em kilobytes (é para isso que serve a opção '-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 > █
```

Neste ponto, você deve estar usando menos de 50 kilobytes no total em sua estrutura de diretório inicial

Se você tiver concluído isso corretamente, seu arquivo `ussout.txt` deve ter:

1. mensagem decodificada secreta, seguida por
2. a saída de seu comando de uso do disco

Verifique o conteúdo de sua saída com `cat ussout.txt`

## 12 FAÇA VALER A PENA

```
/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 >
```

Finalmente, use o redirecionamento para anexar a saída do `date` no mesmo arquivo.

Certifique-se de que **ussout.txt** arquivo agora tem

1. a mensagem secreta decodificada
2. sua saída de uso do disco
3. a saída do date comando

Se tudo parecer bom, envie a tarefa de validação **CHKUSS1** a partir de **ZXP.PUBLIC.JCL** , e se tudo correr bem, você está feito!

Bom trabalho-vamos recapitular	Em seguida ...
<p>Você se conectou ao USS por um shell seguro (SSH), bem como por meio do plug-in do Zowe no VS Code. A partir daí, você navegou em diretórios, gerenciou a saída, hackeou algum texto e aprendeu tudo sobre argumentos. Pode não parecer muito, mas pense no que você fez e no que você sabia quando começou. Se você está na maneira UNIX de fazer as coisas, você provavelmente vai realmente desfrutar do desafio baseado em LINUX que é disponibilizado mais tarde. Se esta não era a sua chávena de chá, pelo menos agora você está feito!</p>	<p>Nós vamos continuar trabalhando no espaço USS para o próximo desafio, então certifique-se de manter esses comandos de terminal e guias VSCode à mão. Nós estaremos fazendo alguma codificação muito básica usando Python. Não entre em pânico se você nunca tiver codificado antes! Vamos mostrar tudo o que você precisa.</p>