

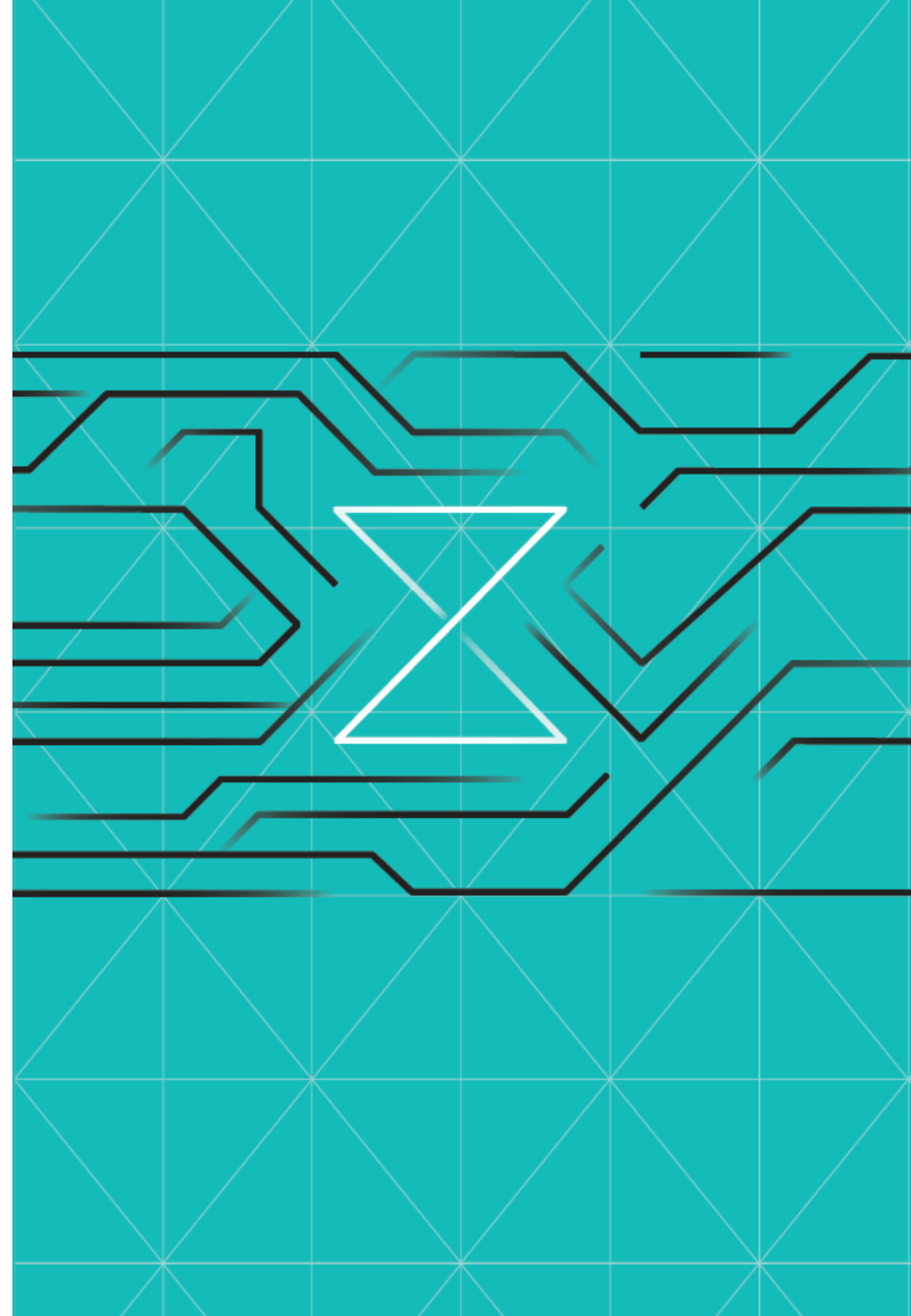
Advanced

230301-1859

USS2

IT'S UNIX. I KNOW THIS!

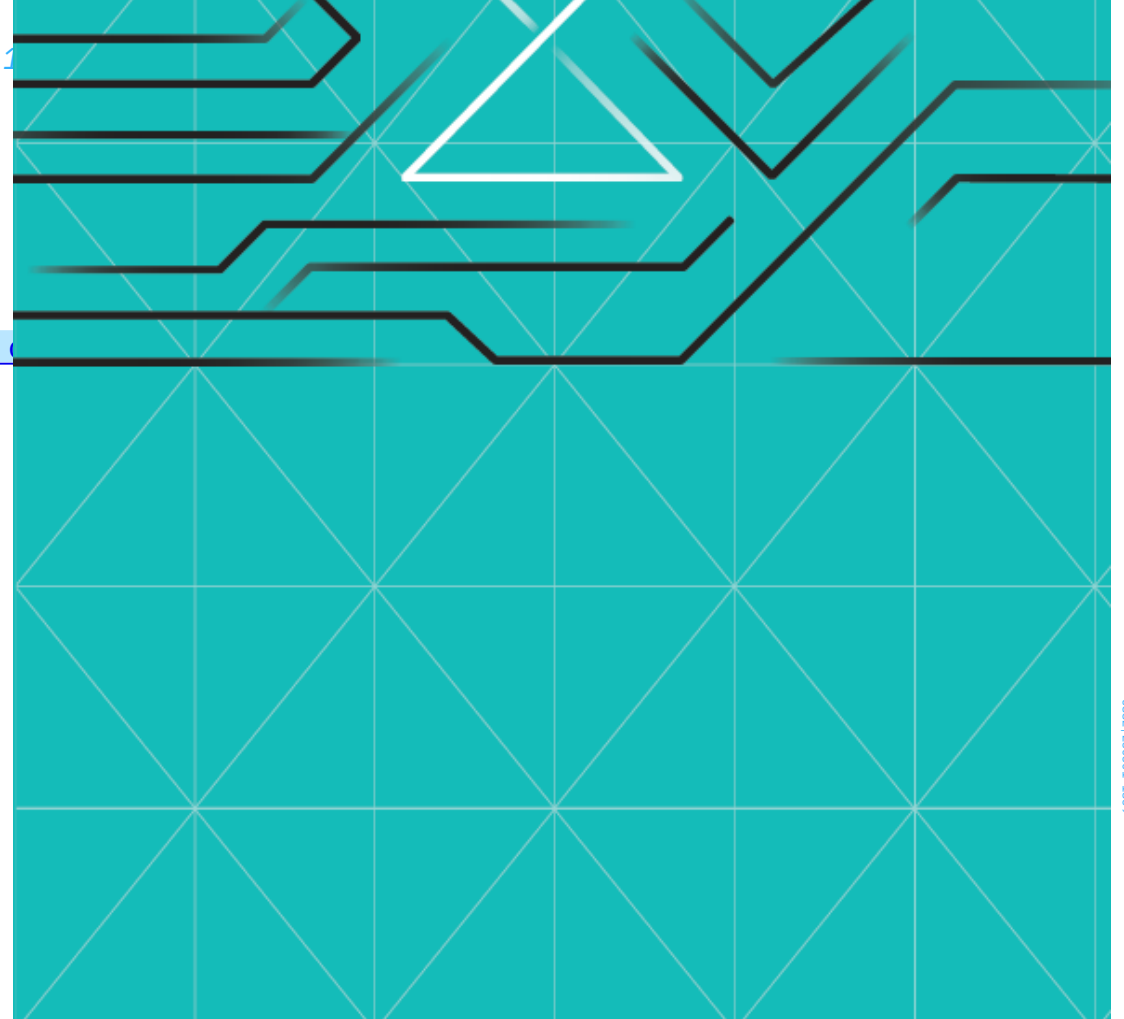
Automated Translation by Watson Language Translator



USS2

IT ' S UNIX. EU SEI DISSO!

- Alguns scripts, alguns em execução, alguns c
 - 0 desafio
 - Antes de começar
 - Investimento
- 1 Um refresco
- 2 Coloque tudo junto
- 3 Localizar e editar animals.sh
- 4 O que está acontecendo aqui?
- 5 corrija a entrada
- 6 fazer o script executar
- 7 executar o script
- 8 verificar conclusão



ALGUNS SCRIPTS, ALGUNS EM EXECUÇÃO, ALGUNS CHMOD'ING

0 Desafio

Existem maneiras de editar arquivos através do terminal, mas para a maioria das pessoas, é muito mais fácil e intuitivo usar o editor de texto construído no VSCode.

Para esse desafio, você examinará o USS usando VSCode, trabalhará em um shell script e o fará executar corretamente para marcar esse desafio como concluído.

Antes De Começar

Você deve ter o VSCode configurado e ter concluído o primeiro desafio do USS no Fundamentals antes de iniciar isso

Além disso, nada mais é necessário, você estará inteiramente no lado USS das coisas.

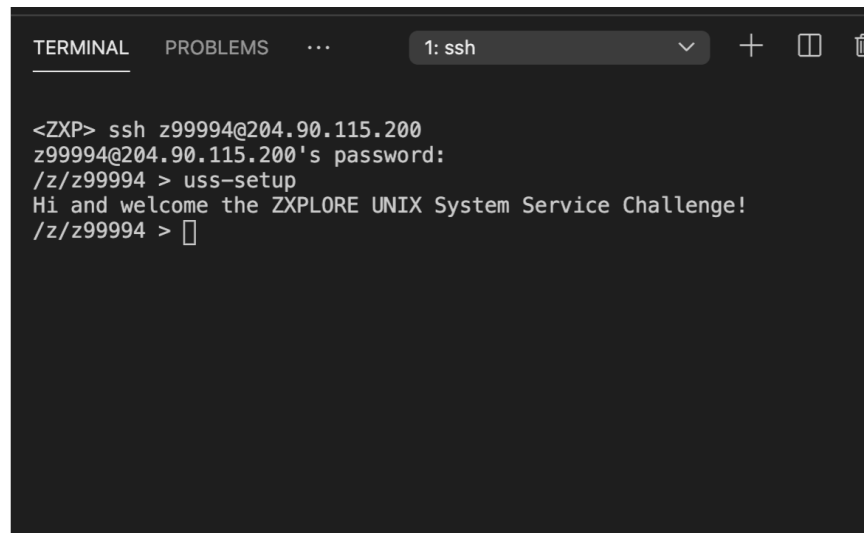
Investimento

Etapas	Duração
8	30 minutos

1 UM REFRESCO

Antes de avançar a todo vapor no desafio USS2, lembre-se de tudo o que você aprendeu.

Primeiro, certifique-se de que seu terminal está aberto, **conectado através de ssh** e configurar corretamente..



```
TERMINAL  PROBLEMS  ...  1: ssh  v  +  []  x

<ZXP> ssh z99994@204.90.115.200
z99994@204.90.115.200's password:
/z/z99994 > uss-setup
Hi and welcome the ZXPLRE UNIX System Service Challenge!
/z/z99994 > 
```

Se você não se lembrar de como fazer isso ou se seu diretório padrão estiver vazio, dê uma olhada na captura de tela acima para localizar a conexão e *uss-configuração* do desafio USS Fundamentals.

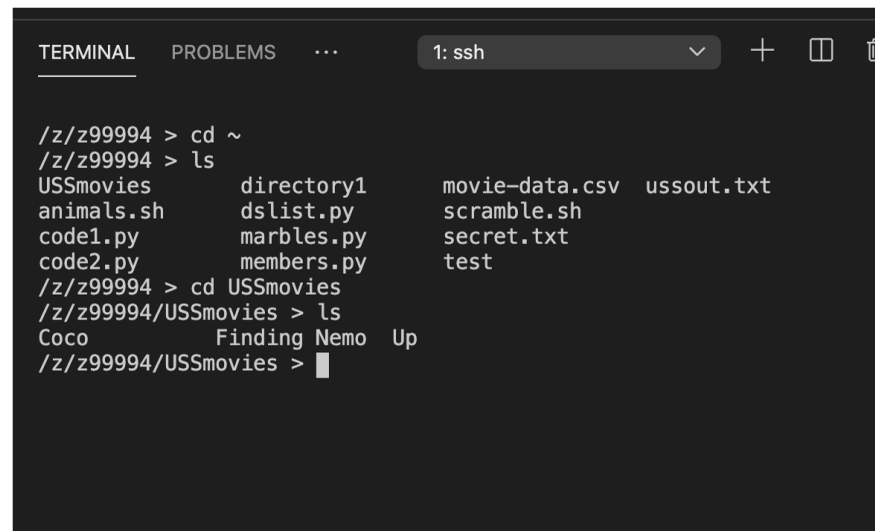
Você sabe como criar um diretório (*mkdir*), crie um arquivo (*toque em*), alterar diretórios (*cd*), olhe em volta (*ls*) e exibindo (imprimir) o diretório ativo (*pwd*).

2 COLOQUE TUDO JUNTO

Uma prática rápida de criar um diretório e colocar arquivos dentro dele.

Volte para o seu diretório inicial no terminal (/z/zxxxxx) e crie um diretório chamado **USSmovies** .

Crie três arquivos em USSmovies nomeados após seus filmes favoritos.

A terminal window with a dark background and light text. The title bar shows '1: ssh'. The terminal content shows a user at the prompt '/z/z99994 >' navigating to the home directory and listing files. The output shows a directory named 'USSmovies' containing various files and subdirectories. The user then navigates into 'USSmovies' and lists its contents, showing files named 'Coco', 'Finding Nemo', and 'Up'.

```
TERMINAL PROBLEMS ... 1: ssh
/z/z99994 > cd ~
/z/z99994 > ls
USSmovies      directory1    movie-data.csv  ussout.txt
animals.sh     dslist.py    scramble.sh
code1.py       marbles.py   secret.txt
code2.py       members.py   test
/z/z99994 > cd USSmovies
/z/z99994/USSmovies > ls
Coco           Finding Nemo  Up
/z/z99994/USSmovies > 
```

Lembrar : Se o nome do seu arquivo tiver espaços, você terá que colocá-lo entre aspas, por exemplo:
`touch "mac and cheese"`.

Se isso não era familiar, continue praticando. É possível continuar incluindo arquivos em seu diretório inicial até que se sinta confortável com os atalhos e a estrutura.

Se você criou com sucesso o diretório e os arquivos, agora é hora de excluí-los.. Você se lembra como fazer isso?

Depois de ter removido com sucesso as informações que você acabou de criar, você não deve mais ver *USSmovies* na listagem de seu diretório inicial

O QUE ESTOU OLHANDO?

Quando você olha em um diretório (ou pasta, se você é um fã do Windows), você precisa de uma ferramenta útil para identificar o que está no diretório, e o que você pode ser capaz de fazer com o conteúdo (entradas).

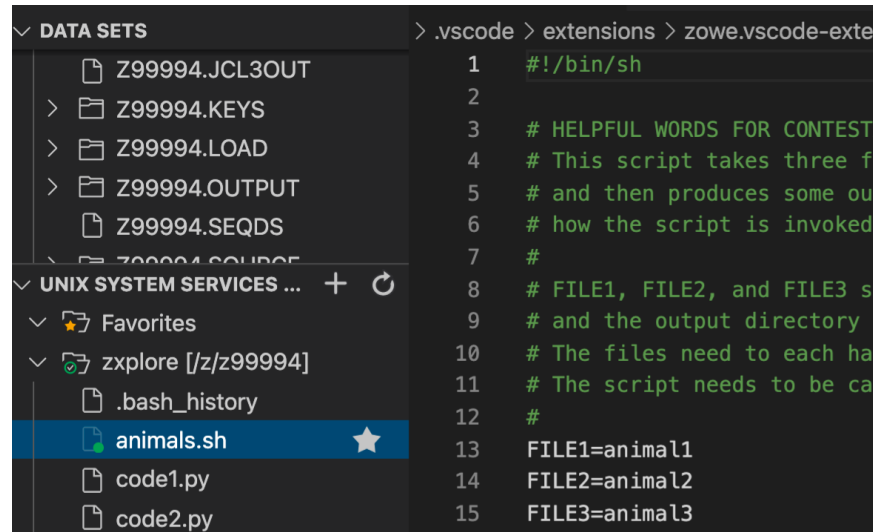
O `ls` em sua forma mais simples mostrará os nomes de arquivos (não ocultos) no diretório, mas ele tem algumas opções comumente usadas para mostrar mais detalhes (confira `man ls` para muito mais informações):

- Eu
mostra as informações do proprietário, o registro de data e hora da última atualização e as permissões
- a
mostra todos os arquivos, incluindo arquivos ocultos (no Unix, nomes de arquivos que começam com "." geralmente não são listados por padrão)
- t
classifica as entradas de diretório por horário da última modificação,
- r
classifica a listagem de entrada de diretório em reverso
- F
inclui um código de caractere após o nome da entrada para mostrar se o arquivo é "especial"

3 LOCALIZAR E EDITAR ANIMALS.SH

Procure em seu diretório inicial e tente localizar **animals.sh** . Isso deveria ter sido copiado no desafio USS Fundamentals, quando você executou o script *uss-configuração* .

Se você não o vir, poderá copiá-lo de **/z/público** (ou execute o script *uss-setup* novamente)



The screenshot shows the IBM Z Xplore interface. On the left, a file explorer pane displays a tree structure. Under 'DATA SETS', there are folders for 'Z99994.JCL3OUT', 'Z99994.KEYS', 'Z99994.LOAD', 'Z99994.OUTPUT', and 'Z99994.SEQDS'. Below this, under 'UNIX SYSTEM SERVICES ...', there is a 'zxplere [/z/z99994]' directory. Inside this directory, the file 'animals.sh' is highlighted with a blue selection bar and a star icon. Other files like '.bash_history', 'code1.py', and 'code2.py' are also visible. On the right, a code editor shows the contents of 'animals.sh'. The script starts with a shebang line '#!/bin/sh' and includes several comments in green text: '# HELPFUL WORDS FOR CONTESTA', '# This script takes three f', '# and then produces some ou', '# how the script is invoked', and '#'. It then defines three variables: 'FILE1=animal1', 'FILE2=animal2', and 'FILE3=animal3'.

Este é um script, exceto que você não vai apenas executar isso imediatamente; você vai corrigi-lo e fazê-lo funcionar.

4 O QUE ESTÁ ACONTECENDO AQUI?

Abra o shell script e olhe ao redor. Há uma grande parte do script que vive dentro de uma instrução "IF", e muito dele só será executado corretamente se certas condições forem atendidas.

```
echo "Checking for the files and folders"
#If there's three files that aren't empty (-s) AND (&&) the directory exist (-d), then do this"
if [ -s "$FILE1" ] && [ -s "$FILE2" ] && [ -s "$FILE3" ] && [ -d "$DIRECTORY1" ]; then
    echo "Everything looks good"

    #Copy the three files into that directory
    cp $FILE1 $FILE2 $FILE3 $DIRECTORY1
    echo "Files have been copied!"

    #Produce this output and put it in the "message" file
    echo "We're extremely happy to have $USERNAME on the system" > "$DIRECTORY1"/message
    MODELTYPE=$(uname -m)
    OS=$(uname -I)
    echo "The operating system is $OS running on a model type $MODELTYPE" >> "$DIRECTORY1"/message
    echo "" >> "$DIRECTORY1"/message
    echo "If $USERNAME looks out the window, they might say:" >> "$DIRECTORY1"/message

    #For every file (there should be three), do this
    for file in "$DIRECTORY1"/animal*
    do
        CURRENT_ANIMAL=$(cat $file)
        echo "Why hello there, $CURRENT_ANIMAL" >> "$DIRECTORY1"/message
    done
```

Leia os comentários para obter informações adicionais e tente descobrir o que está acontecendo no script antes de continuar.

5 CORRIJA A ENTRADA

O script tem muitas pistas sobre o que precisa ser executado.

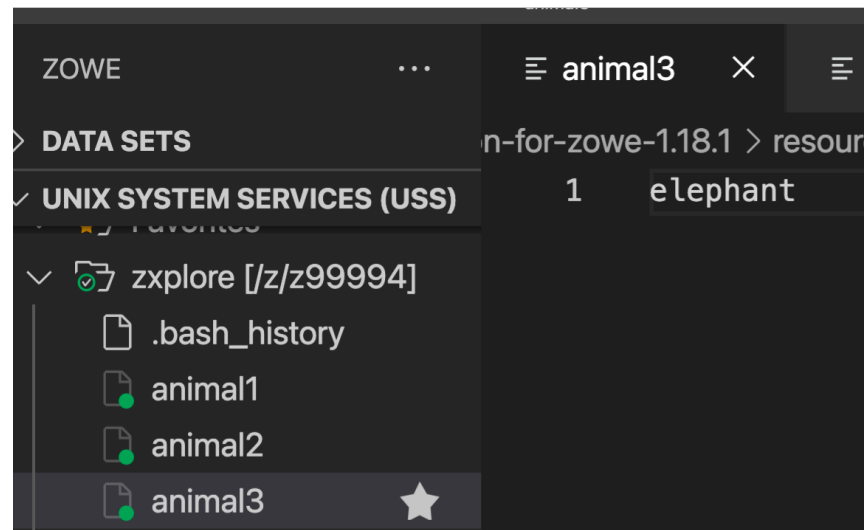
Criar três arquivos (*animal1* , *animal2* , e *animal3*) e um diretório abaixo de seu diretório inicial chamado **uss2output**

Use o VSCode para abrir cada um desses arquivos e inserir uma única linha em cada um deles, simplesmente nomeando um animal que você provavelmente verá em sua área.

Não pense muito nisso, mas certifique-se de colocar algo lá, caso contrário, o script não pode funcionar.

Lembre-se de completar cada linha com um Enter ou Return para que seja uma "linha", não apenas uma "palavra"!

(Como um teste simples, `cat animal[123]` deve mostrar 3 linhas)



Observe a captura de tela acima O arquivo *animal1* precisa ter um animal nomeado nele. O mesmo é verdadeiro para *animal2* e *animal3*

Conforme você trabalha entre o terminal e a GUI, pode haver momentos em que as visualizações se tornam inconsistentes.

Lembre-se de que você pode precisar ocasionalmente recarregar a visualização do VSCode para capturar as atualizações mais recentes.

"O MUNDO PRECISA DE OUTRA LINGUAGEM DE SCRIPT?"

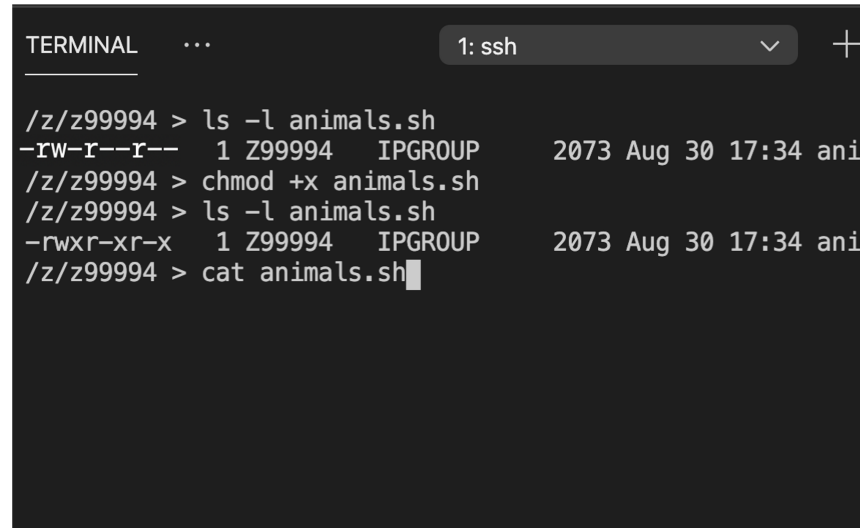
Um número crescente de programas em execução no Z veio do Linux ou de outras plataformas UNIX, onde os shell scripts são usados para executar de forma simples e transparente as etapas de configuração, como verificar se há espaço em disco suficiente, criar diretórios de destino e fazer. Certifique-se de que o software de pré-requisito esteja instalado.

Uma das razões pelas quais as pessoas realmente gostam de scripts no UNIX é que os comandos são os mesmos que você pode usar ao digitar em um shell no modo interativo, mas com o benefício adicional de adicionar as instruções "IF" e "FOR".

O **sh** shell no USS é um pouco limitado, mas você pode usar o mais completo **bash** shell mais tarde se você gostaria de tentar algo mais sofisticado.

6 FAZER O SCRIPT EXECUTAR

Volte para sua sessão de terminal (`ssh`) e executar `ls -l animals.sh`.



```
TERMINAL  ...  1: ssh  +
/z/z99994 > ls -l animals.sh
-rw-r--r--  1 Z99994  IPGROUP      2073 Aug 30 17:34 ani
/z/z99994 > chmod +x animals.sh
/z/z99994 > ls -l animals.sh
-rwxr-xr-x  1 Z99994  IPGROUP      2073 Aug 30 17:34 ani
/z/z99994 > cat animals.sh
```

Você deve ver o arquivo mostra as permissões `_r_ead` e `_w_rite`, mas nenhuma permissão `e_x_ecution` Em seguida, torne o arquivo executável, emitindo o comando `_ch_ange _mod_e`: `chmod +x animals.sh`

Executar `ls -l animals.sh` novamente, e ver se você pode ver a diferença.

Há um **x** adicionado às permissões, deixando você saber que ele é executável, o que significa que ele pode ser executado como um programa.

R significa Leitura, W significa Gravação e X significa Execução.

Mais um comando rápido: o `cat` comando (abreviação para `con_cat_enate`) envia o conteúdo de um arquivo para a tela. Use isso para espiar o conteúdo de um arquivo que você não precisa editar.

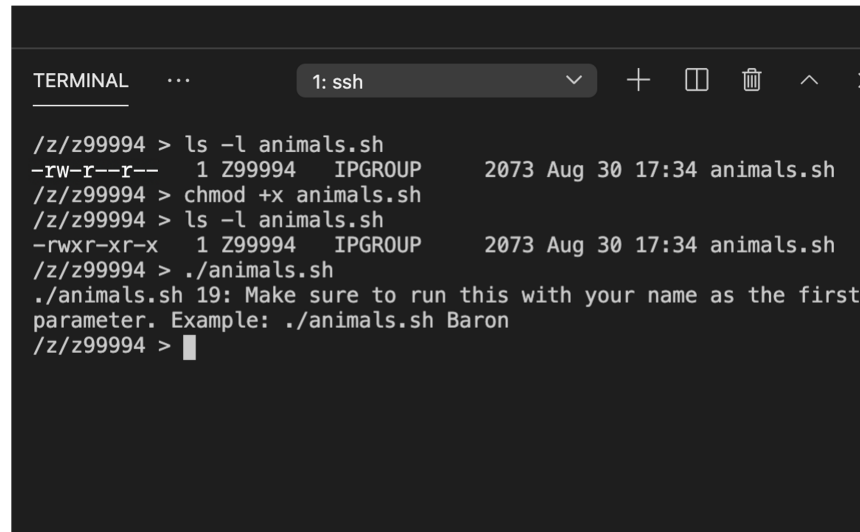
Exemplo: `cat animals.sh`

(*mais* exibe arquivos também, mas mostra uma tela por vez)

7 EXECUTAR O SCRIPT

Tipo `./animals.sh`

No UNIX, quando você fornece apenas o caminho para um arquivo, assume-se que você deseja executá-lo. (Lembre-se de usar `cat` se você deseja exibiá-la, não execute-a)

A terminal window titled '1: ssh' showing a series of commands and their outputs. The user is at the prompt '/z/z99994 >'. They run 'ls -l animals.sh', which shows permissions '-rw-r--r--' and owner '1 Z99994 IPGROUP'. Then they run 'chmod +x animals.sh', which changes the permissions to '-rwxr-xr-x'. Finally, they run './animals.sh', which outputs a message: './animals.sh 19: Make sure to run this with your name as the first parameter. Example: ./animals.sh Baron'. The prompt returns to '/z/z99994 >'.

```
TERMINAL ... 1: ssh
/z/z99994 > ls -l animals.sh
-rw-r--r--  1 Z99994  IPGROUP    2073 Aug 30 17:34 animals.sh
/z/z99994 > chmod +x animals.sh
/z/z99994 > ls -l animals.sh
-rwxr-xr-x  1 Z99994  IPGROUP    2073 Aug 30 17:34 animals.sh
/z/z99994 > ./animals.sh
./animals.sh 19: Make sure to run this with your name as the first
parameter. Example: ./animals.sh Baron
/z/z99994 > █
```

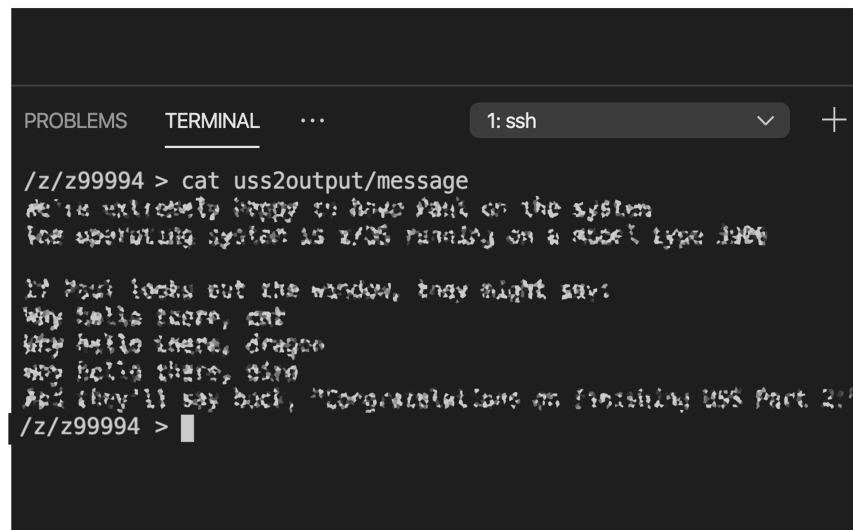
Em vez de digitar o caminho completo para o script, você pode apenas usar um único ponto e uma barra para dizer "A partir deste diretório atual", em seguida, o arquivo de script que você deseja executar.

O script será executado, mas ele se queixará de um primeiro parâmetro ausente. Siga seu exemplo com seu nome e tente novamente.

8 VERIFICAR CONCLUSÃO

Execute o comando *animals.sh* script novamente, desta vez dando a ele um argumento (a parte que vem após o próprio script).

O próprio script diz o que isso deve ser, juntamente com tudo o mais que você precisa para fazer essa execução corretamente.



```
PROBLEMS  TERMINAL  ...  1: ssh  v  +
/z/z99994 > cat uss2output/message
We're extremely happy to have Paul on the system
The operating system is z/OS running on a model type 1400

If Paul looks out the window, they might say:
Why hello there, cat
Why hello there, dragon
Why hello there, bird
And they'll say back, "Congratulations on finishing USS Part 2!"
/z/z99994 > 
```

Quando tiver concluído, efetue o registro de saída em **~/uss2output/mensagem** e você deve ver uma mensagem muito claramente felicitando você por completar sua tarefa.

Parece bem? Se sentindo bem? Muito bem.

Localize o **CHKUSS2** tarefa em **ZXP.PUBLIC.JCL** , clique com o botão direito sobre ele e, em seguida, selecione "Enviar Tarefa" para marcar este desafio como concluído.

Bom trabalho-vamos recapitular	Em seguida ...
<p>Você pegou um script, descobriu o que ele faria e o fez funcionar. Trabalhar no UNIX System Services significa que você provavelmente estará fazendo esse tipo de coisa muito, então é bom que você foi capaz de descobrir.</p> <p>Fizemos o script cheio de exemplos de propósito, já que você pode querer emprestar algumas dessas expressões e verificações condicionais mais tarde para suas próprias criações, ou envios de concurso. (dica, dica!)</p>	<p>Se você já fez os desafios PDS e JCL2, sua próxima tarefa é ir atrás desses desafios ZOAU e Ansible. Eles usam tudo o que você aprendeu até agora e o reúnem de uma maneira única que realmente permite que você mostre suas habilidades.</p>