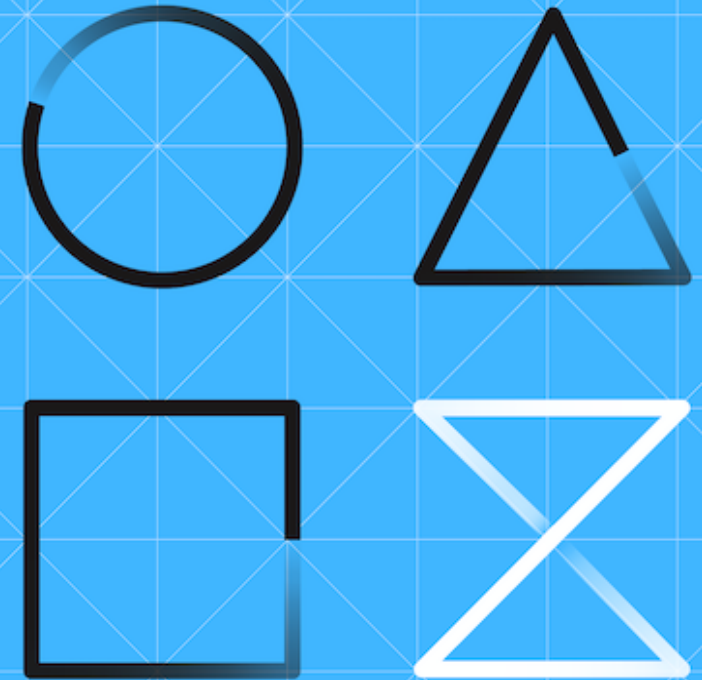


REXX1

Enterprise IT Duct Tape

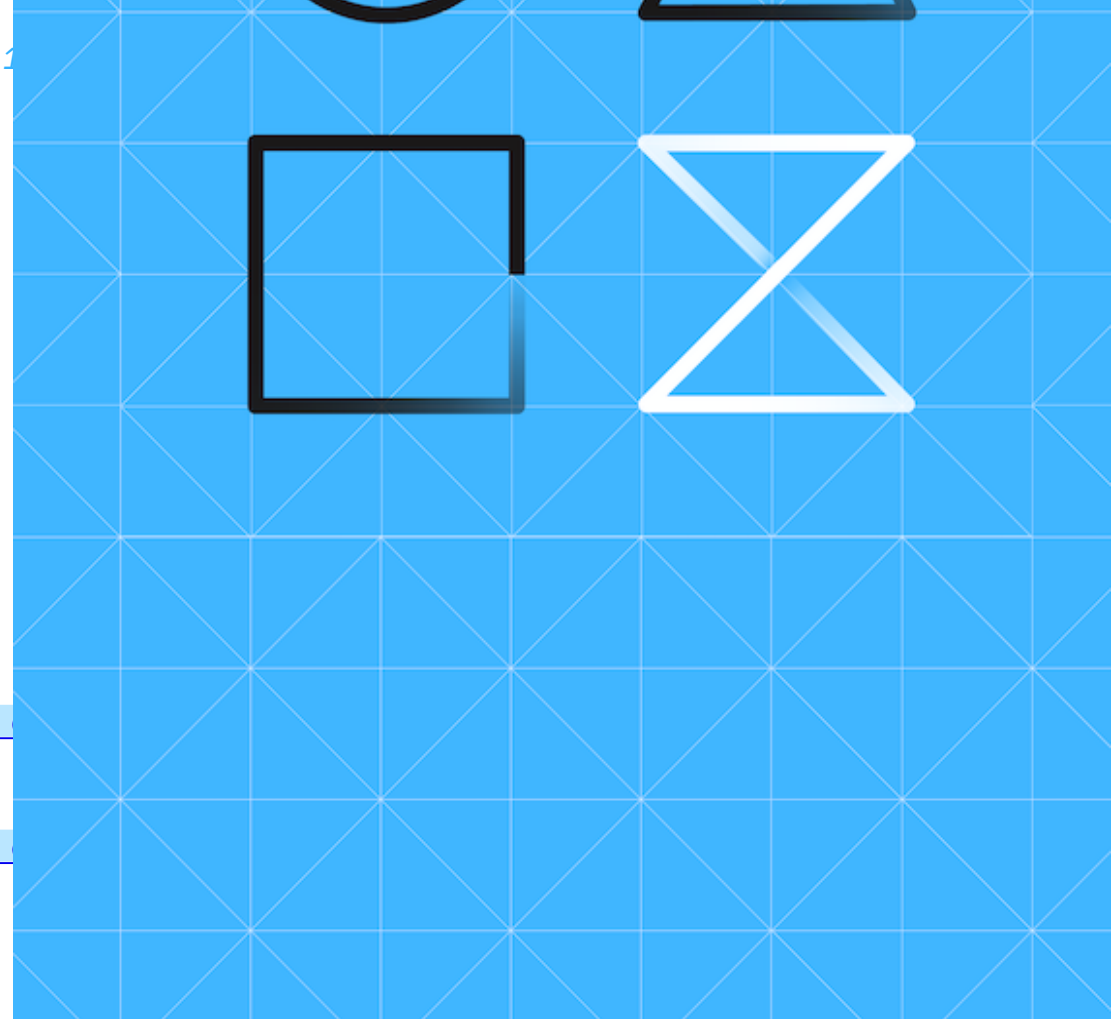
Automated Translation by Watson Language Translator



REXX1

Fita de Duc de TI Corporativa

- [Fita de Duc de TI Corporativa](#)
 - [0 desafio](#)
 - [Antes de começar](#)
 - [Investimento](#)
- [1 Instalar o zowe cli](#)
- [2 Instalação do ZOWE NPM em MacOS](#)
- [3 Instalação do ZOWE NPM para Windows](#)
- [4 origem e terminal](#)
- [5 criar um perfil do tso](#)
 - [5.1 Usando arquivos de configuração de](#)
 - [5.2 Usando perfis de Conexão \(V1\)](#)
- [6 configurar perfis padrão](#)
 - [6.1 Usando arquivos de configuração de](#)
 - [6.2 Usando perfis de Conexão \(V1\)](#)
 - [6.3 Teste sua conexão padrão](#)
- [7 Execute seu primeiro REXX](#)
- [8 INICIAR UM ESPAÇO DE ENDEREÇO](#)
- [9 Executar o mesmo rexx](#)
- [10 Adivinha o quê? outro executável](#)
- [11 Veja como o código é bonito](#)
- [12 Bate-papo ao vivo com REXX](#)



REXX1|240416-1816

FITA DE DUC DE TI CORPORATIVA

0 Desafio

Neste desafio, você encontrará o Rexx ("Executor Estendido Reestruturado"), uma linguagem de programação conhecida por sua simplicidade, potência e facilidade relativa de uso. Você vai mergulhar em como executar um programa Rexx a partir de uma linha de comando, bem como como iniciar um TSO 'Address Space' para executar um programa Rexx interativo.

Para interagir com o TSO, você precisará da interface da linha de comandos (CLI) do ZOWE instalada

Antes De Começar

Esse desafio requer alguma configuração para o seu ambiente de terminal

Certifique-se de executar as etapas de instalação da CLI do ZOWE.

Investimento

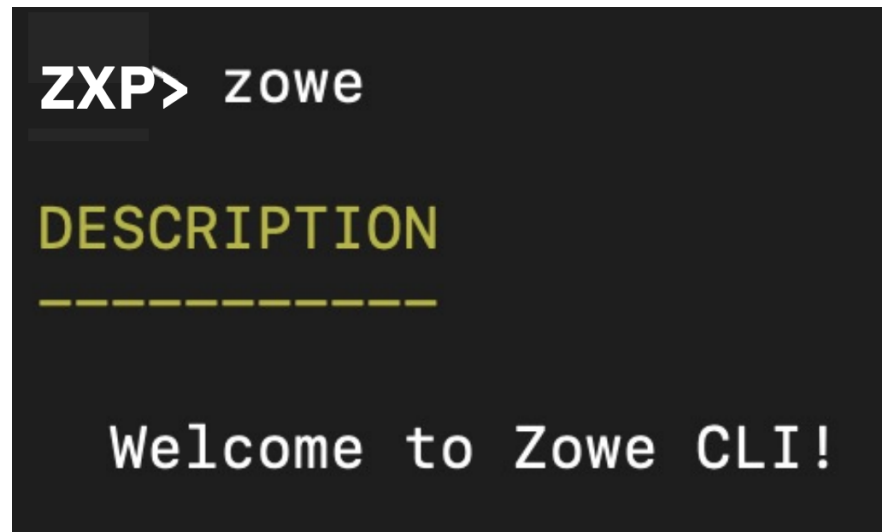
Etapas	Duração
12	60 minutos

REXX|240416-1816

1 INSTALAR O ZOWE CLI

Até agora, você tem usado o plug-in do Zowe Explorer para VSCode, mas o Zowe pode fazer muito, muito mais e é responsável por trazer muito mais para o mainframe.

Para ser claro, você estará instalando a interface da linha de comando (CLI) do Zowe em seu próprio computador, *não* no mainframe.



Você usará a CLI do Zowe para trabalhar com o serviço z/OSMF que está em execução no mainframe, mas você estará conduzindo a maior parte desse desafio a partir do seu próprio computador

Os usuários do Linux podem precisar fazer um pouco de exploração para encontrar o que funciona em seu sistema específico, mas ele deve olhar mais perto das etapas do Mac, apenas substituindo seu arquivo de perfil shell correto.

Os usuários do Windows podem precisar modificar a variável de ambiente PATH do usuário para serem capazes de executar com êxito `zowe` como um comando-mais sobre isso mais tarde.

2 INSTALAÇÃO DO ZOWE NPM EM MACOS

Como a CLI do Zowe usa pacotes de nós, ela requer que uma versão suportada de [Node.js](#) está instalado e adequadamente acessível

Para usar pacotes de nós no sistema operacional, você precisará carregá-los em um diretório global '.npm' que pode ser acessado por usuários regulares.

As etapas a seguir configurarão isso, informarão ao npm (o Node Package Manager) para usá-lo e incluirão isso na lista normal de locais em que o sistema operacional procura programas para execução.

```
ZXP>mkdir ~/.npm-global
ZXP> npm config set prefix '~/.npm-global'
echo "export PATH=~/.npm-global/bin/:$PATH" >> .zprofile
source .zprofile
npm i -g @zowe/cli
ZXP>echo "export PATH=~/.npm-global/bin/:$PATH" >> .zprofile
ZXP>source .zprofile
ZXP>npm i -g @zowe/cli
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported

> @zowe/cli@6.33.0 preinstall /Users/rcruicks/.npm-global/lib/node_modules/@zowe/cli
> node ./scripts/preinstall

/Users/rcruicks/.npm-global/bin/bright -> /Users/rcruicks/.npm-global/lib/node_modules/@zowe/cli/lib/main.js
/Users/rcruicks/.npm-global/bin/zowe -> /Users/rcruicks/.npm-global/lib/node_modules/@zowe/cli/lib/main.js

> @zowe/cli@6.33.0 postinstall /Users/rcruicks/.npm-global/lib/node_modules/@zowe/cli
> node ./scripts/validatePlugins

Since you re-installed Zowe CLI, we are re-validating any plugins.
No plugins have been installed into your CLI application.
+ @zowe/cli@6.33.0
added 287 packages from 202 contributors in 19.319s
ZXP>
```

Para a maioria dos usuários do MacOS, eles devem fazer o truque..

1. mkdir ~/.npm-global
2. npm config set prefix ~/.npm-global
3. echo "PATH=~/.npm-global/bin/:\$PATH" >> .zprofile
4. echo "export PATH" >> .zprofile
5. source .zprofile
6. npm i -g @zowe/cli

Nota: se o shell do terminal padrão for bash em vez de Zsh, essas instruções precisarão ser modificadas para atualizar seu `.bashrc` ou `.bash_profile` em vez do arquivo `.zprofile`.

3 INSTALAÇÃO DO ZOWE NPM PARA WINDOWS

Em seu laptop ou estação de trabalho Windows, primeiro alterne para um terminal CMD em vez do terminal PowerShell padrão -em seguida, instale a CLI do Zowe usando 'npm', o Node Package Manager.

Isso deve funcionar para a maioria dos usuários, embora sua saída possa ser diferente do que você vê na captura de tela.

1. tipo cmd (isso mudará o shell para CMD do PowerShell)
2. npm i -g @zowe/cli
3. zowe

```
PS C:\Users\JeffreyBisti> cmd
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\JeffreyBisti>npm i -g @zowe/cli
C:\Users\JeffreyBisti\AppData\Roaming\npm\bright -> C:\Users\JeffreyBisti\AppData\Roaming\np
e_modules\@zowe\cli\lib\main.js
C:\Users\JeffreyBisti\AppData\Roaming\npm\zowe -> C:\Users\JeffreyBisti\AppData\Roaming\npm\
modules\@zowe\cli\lib\main.js

> @zowe/cli@6.22.0 postinstall C:\Users\JeffreyBisti\AppData\Roaming\npm\node_modules\@zowe\
> node ./scripts/validatePlugins

Since you re-installed Zowe CLI, we are re-validating any plugins.
No plugins have been installed into your CLI application.
+ @zowe/cli@6.22.0
updated 4 packages in 14.432s

C:\Users\JeffreyBisti>zowe
```

Se você vir uma resposta como "zowe: command not found", feche a janela do terminal e ative-a novamente. Lembre-se de alternar para o CMD.

Se você ainda vir um erro "comando não encontrado", você precisará atualizar sua variável de ambiente PATH para incluir o local do programa zowe-geralmente em uma pasta "node_modules/bin" em C:, ou em seu diretório inicial.

Também é possível tentar executar o segundo comando novamente, mas remover a opção '-g':

```
npm i @zowe/cli
```

Execute uma procura na Internet para saber como configurar sua variável PATH do Windows, pois ela varia de versão para versão e também pode ser diferente se seu laptop tiver sido configurado para perfis móveis e políticas de grupo.

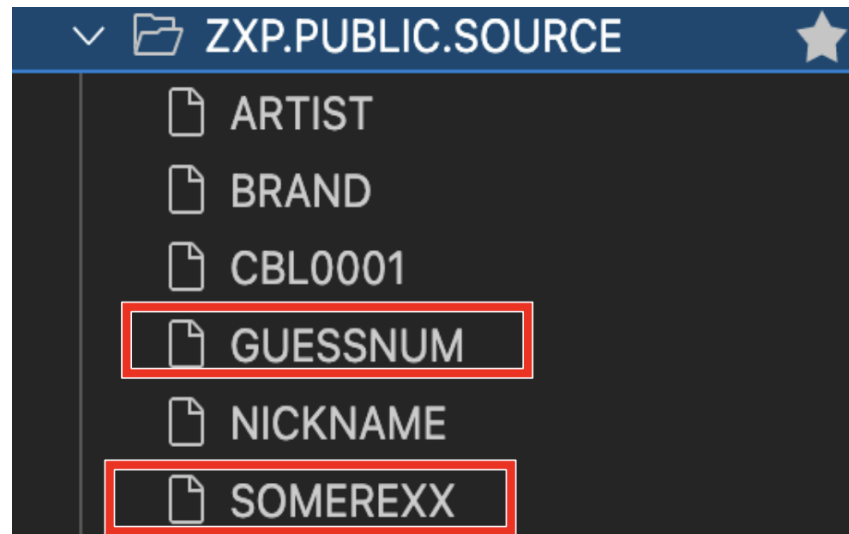
Em geral, se comandos de código aberto como zowe e ssh não funcionarem conforme esperado, tente executar o comando a partir de um terminal CMD, em vez de um terminal PowerShell. Se o prompt da janela do terminal no VSCode começar com `PS >` ou semelhante, tipo `CMD` para chegar a um ambiente "shell" mais fácil.

REXX1240416-1816

4 ORIGEM E TERMINAL

Comece sua viagem de programação REXX copiando dois membros de ZXP.PUBLIC.SOURCE em seu próprio conjunto de dados SOURCE.

Especificamente, você está procurando 'SOMEREXX' e 'GUESSNUM'



Em seguida, abra uma janela do Terminal, dentro do VSCode ou diretamente do seu sistema operacional de estação de trabalho / laptop, assim como você fez para o desafio USS, mas *NÃO* use SSH para se conectar ao mainframe desta vez.

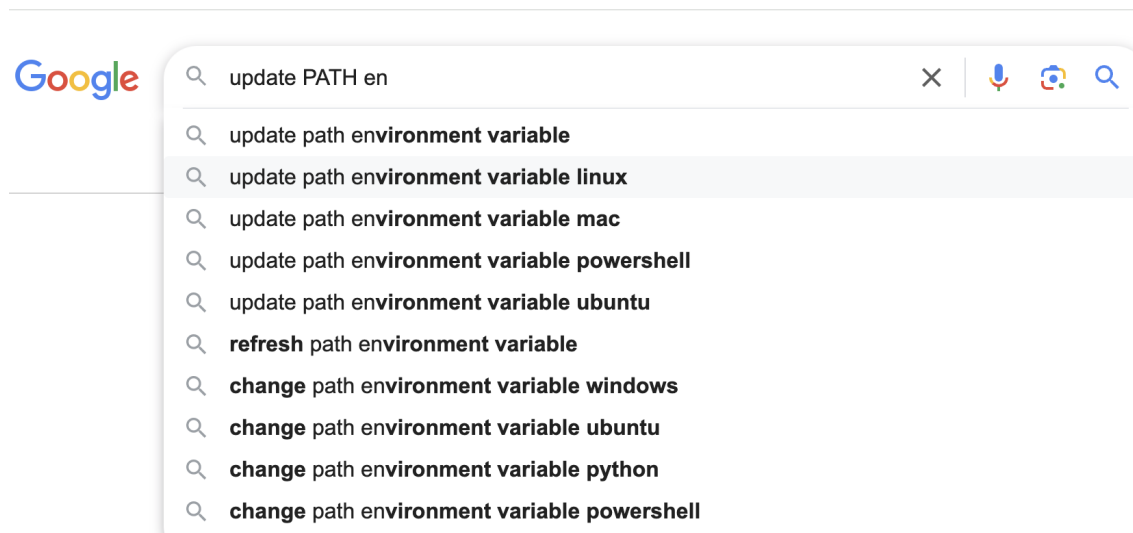
Você deve ser capaz de digitar o comando `zowe` a partir daqui e obter alguma saída útil do programa:

DESCRIPTION

Welcome to Zowe CLI!

Zowe CLI is a command line interface (CLI) that provides a simple and streamlined way to interact with IBM z/OS.

Se você vir uma mensagem como "comando não localizado", será necessário localizar onde o comando zowe está instalado e incluir esse local em sua variável de ambiente PATH. Como isso é feito depende do sistema operacional que você está usando. Use a procura na internet para obter instruções que melhor correspondam ao seu laptop/estação de trabalho::



REXX|240416-1816

5 CRIAR UM PERFIL DO TSO

Até agora, no VSCode, você está usando um perfil z/OSMF para fazer conexões com o mainframe

Agora também pode ser necessário criar um perfil TSO do zowe para emitir comandos TSO usando a CLI do Zowe.

5.1 USANDO ARQUIVOS DE CONFIGURAÇÃO DE EQUIPE (V2)

Insira o seguinte comando para ver quais perfis já estão definidos:

```
zowe config list
```

Você deve ver uma impressão que mostre seu perfil do zOSMF:

```
zosmfZXPlore:
  type:      zosmf
  properties:
    port:      10443
    host:      204.90.115.200
    rejectUnauthorized: false
    user:      (secure value)
    password:  (secure value)
  secure:
    - user
    - password
```

e o perfil básico do ORT:

```
tsoZXPlore:
  type:      tso
```

- properties:
 - account: FB3
 - codePage: 1047
 - logonProcedure: IZUFPROC
- secure:
 - (empty array)

Também deve haver seção no final da impressão que mostra os perfis padrão:

- defaults:
 - zosmf: zosmfZXPLore
 - tso: tsoZXPLore

Seus nomes de perfil podem ser diferentes-certifique-se de que os nomes no `defaults` seção corresponder aos seus perfis.

5.2 USANDO PERFIS DE CONEXÃO (V1)

Se você originalmente criar sua conexão VSCode com o mainframe usando um "Perfil de conexão" em vez de uma "Configuração de equipe", as informações a seguir se aplicarão:

```
OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS

Jxxxxx@xxxxx$ zowe profiles create tso-profile Zxplore -a FB3
Profile created successfully! Path:
/Users/xxxxx.zowe/profiles/tso/Zxplore.yaml

account:      FB3
characterSet: 697
codePage:     1047
columns:      80
logonProcedure: IZUFPROC
regionSize:   4096
rows:         24

Review the created profile and edit if necessary using the profile update command.
Jxxxxx@xxxxx$
```

Digite este comando:

```
● zowe profiles create tso-profile zxplore -a FB3
```

Use qualquer nome de perfil que você quiser-'zxplore ' é apenas o nosso exemplo e se você tem seguido todos os nossos exemplos, é bom manter-se consistente.

(Observe que a captura de tela mostra o usuário criando um perfil 'Zxplore'-isso criaria um perfil separado de 'zxplore ')

6 CONFIGURAR PERFIS PADRÃO

Neste ponto, é necessário certificar-se de que os perfis corretos sejam configurados como padrões

Isso significa que quando você emite um comando da CLI zowe, você não deve ter que fornecer o nome do host, a porta, o ID do usuário, a senha, etc. quando o comando é executado como todas essas informações estarão disponíveis a partir dos perfis.

6.1 USANDO ARQUIVOS DE CONFIGURAÇÃO DE EQUIPE (V2)

Seus padrões já devem estar definidos, mas você pode verificar usando:

```
● zowe config list defaults
```

que deve mostrar algo semelhante a:

```
zosmf: zosmfZXPlore
tso:   tsoZXPlore
ssh:   ssh
```

6.2 USANDO PERFIS DE CONEXÃO (V1)

Eles provavelmente já são seus padrões, mas apenas no caso de:

```
zowe profiles set-default zosmf zxplore
```

```
zowe profiles set-default tso zxplore
```

(É claro, use os nomes de perfil que você escolheu se você usou nomes diferentes de 'zxplore ').

```

/Users/joao/.zowe/profiles/tso/Zxplore.yaml

account:      FB3
characterSet: 697
codePage:     1047
columns:      80
logonProcedure: IZUFPROC
regionSize:   4096
rows:         24

Review the created profile and edit if necessary using the p
joao@joao-laptop:~$ zowe profiles set-default zosmf
The default profile for zosmf set to Zxplore
joao@joao-laptop:~$ zowe profiles set-default tso ;
The default profile for tso set to Zxplore
joao@joao-laptop:~$

```

Se você quiser ou precisar começar de novo, use o `zowe profiles delete` e siga os avisos.

É possível usar o comando `zowe profiles list zosmf` para mostrar seu perfil zosmf padrão atual. (procure '(default)' na saída)

Se você tiver outros perfis de outras atividades ou conexões de mainframe, às vezes uma reinicialização do VSCode será necessária para que um comutador de perfil entre em vigor.

6.3 TESTE SUA CONEXÃO PADRÃO

Um teste rápido para certificar-se de que seus perfis padrão estejam funcionando-execute um comando simples da lista de conjuntos de dados:

● `zowe files list ds "ZXP.PUBLIC.*"`

isto deve responder com:

```

ZXP.PUBLIC.CLIST
ZXP.PUBLIC.DATA
ZXP.PUBLIC.DB2.CBL

```

```
ZXP.PUBLIC.DB2.DATA  
ZXP.PUBLIC.DB2.JCL  
ZXP.PUBLIC.DB2.SQL  
ZXP.PUBLIC.EXEC  
ZXP.PUBLIC.GLOBAL.SMPPTS  
ZXP.PUBLIC.INPUT  
ZXP.PUBLIC.JCL  
...
```

Se, em vez disso, você for solicitado para qualquer opção ausente-host, porta, número da conta, etc-então seus perfis de padrões não estão (ainda) configurados corretamente. Revise as últimas seções e certifique-se de que todas as configurações necessárias estejam em vigor.

7 EXECUTE SEU PRIMEIRO REXX

Digite o seguinte comando:

```
zowe tso issue command "exec 'Zxxxxx.SOURCE(somerexx)'" --ssm
```

Deve-se receber uma mensagem de saudação do mainframe.



Certifique-se de incluir todas as aspas duplas e simples.

Se você receber solicitações para 'account', 'hostname', 'userid' ou 'password', você quase certamente não configurou os perfis padrão corretamente-retorne algumas etapas e corrija conforme necessário.

Esta pode ser a última vez que apontamos que sempre que você vê Zxxxxxx ou Z99994, você precisa inserir seu próprio ID do usuário.

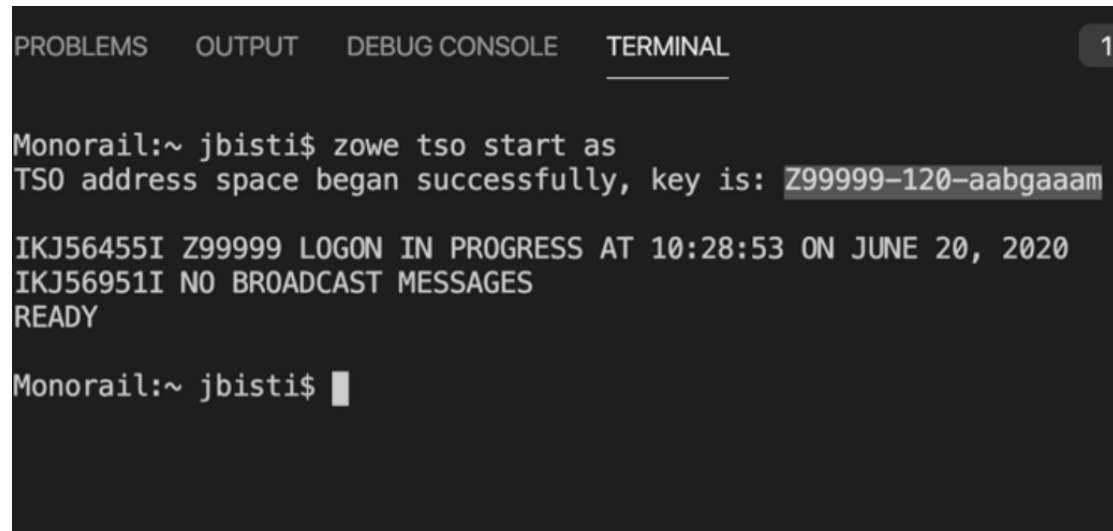
8 INICIAR UM ESPAÇO DE ENDEREÇO

Desta vez, em vez de um único comando, você executará um programa REXX interativo; para fazer isso funcionar, será necessário criar um 'espaço de endereço' do servidor para iniciar e executar o comando REXX até concluir.

Inicie um espaço de endereço com o comando a seguir:

```
zowe tso start as
```

(aqui 'as' é abreviado para 'address-space')



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  1:

Monorail:~ jbisti$ zowe tso start as
TSO address space began successfully, key is: Z99999-120-aabgaaam

IKJ56455I Z99999 LOGON IN PROGRESS AT 10:28:53 ON JUNE 20, 2020
IKJ56951I NO BROADCAST MESSAGES
READY

Monorail:~ jbisti$
```

Isso criará um espaço de endereço para você e dirá a você seu **chave** , que começará com seu ID do usuário (como você pode ver acima).

Você precisará dessa chave para as próximas etapas-ela será referida como "{my-as-key}".

Às vezes, depois de não ser usado por um tempo, um espaço de endereço do TSO desaparece. Se isso acontecer, basta fazer outro usando o mesmo comando start command-mas tome nota da nova chave que ele retorna.

É possível optar por parar um espaço de endereço com o comando:

```
zowe tso stop as {my-as-key}
```

9 EXECUTAR O MESMO REXX

```
Monorail:~ jbisti$ zowe tso send as Z99999-120-aabgaaam --data "exec 'Z99999.source(somerexx)'"
GREETINGS
FROM
REXX
READY
Monorail:~ jbisti$
```

Execute o mesmo programa REXX a partir da etapa 7, mas desta vez, direcione a entrada para o espaço de endereço (e lembre-se, você provavelmente pode pressionar a seta para cima para rechamar comandos anteriores)

```
zowe tso send as {my-as-key} --data "exec 'Zxxxxx.SOURCE(somerexx)'"
```

Notas :

1. Esse comando está tudo em uma linha
2. {my-as-key} é a chave do espaço de endereço que você acabou de iniciar
3. desta vez, deixe fora a opção '-ssm'

Você deve voltar exatamente a mesma resposta de antes.

A grande diferença aqui é que agora você está emitindo esses comandos para um espaço de endereço de TS0 semi-persistente, que fará mais sentido na próxima etapa.

" TSO? ESPAÇO DE ENDEREÇO? "

TSO (Time Sharing Option) é outra maneira que o z/OS permite que muitos usuários obtenham acesso a conjuntos de dados, executem programas e visualizem a saída. É essencialmente a interface da linha de comandos para z/OS (quando você não estiver usando USS para acessar o lado UNIX das coisas).

Pense em um 'Address Space' como um ticket que permite que você comece a usar a memória do sistema. Um espaço de endereço representa uma enorme quantidade de memória, embora o sistema ainda irá controlar o que vive na memória real no chip, versus o que é movido (ou paginado) para o disco.

Para onde seu programa vai sua memória depende de quão importante ele é, como ele deve ser usado e se ele será compartilhado com outros programas.

Os Espaços de Endereço são uma parte principal do z/OS e você deve ler mais sobre eles quando tiver um minuto:

https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zconcepts/zconcepts_82.htm

10 ADIVINHA O QUÊ? OUTRO EXECUTÁVEL

Supondo que seu espaço de endereço TSO esteja ativo, execute o programa **GUESSNUM** usando o mesmo comando (basta alterar o nome do membro de SOMEREXX para GUESSNUM)...

```
READY  
Monorail:~ jbsti$ zowe tso send as Z99999-120-aabgaaam --data "exec 'Z99999.source(guessnum)'"  
I'm thinking of a number between 1 and 10.  
What is your guess?
```

Este é um programa que gera um número aleatório entre 1 e 10, e você tem que adivinhar esse número.

Pode haver outros jogos que você prefere jogar agora, mas nem todos eles vão te ensinar sobre REXX e TSO, então tenha isso em mente se você estiver olhando para o seu XBOX!

REXX|240416-1816

11 VEJA COMO O CÓDIGO É BONITO

Se você ainda não tiver feito isso, abra o código para esse programa no editor VSCode.

Não é um programa complicado por qualquer meio, mas você pode estar percebendo o quão simples este código REXX realmente é.

16 linhas de código, incluindo um comentário e uma linha em branco; e sim, aqueles comandos "say" e "pull" realmente fazem o que você acha que eles fazem.

Você pode ver por que as pessoas amam essa linguagem.

Também tem o que eu considero ser o maior logotipo do mundo para uma linguagem de programação.

Olhe para ele. Apenas. Olha só. Em. Isso ...



12 BATE-PAPO AO VIVO COM REXX

Agora você pode enviar suas suposições para o programa, substituindo tudo entre as aspas duplas por um número.

```
Monorail:~ jbisti$ zowe tso send as Z99999-120-aabgaaam --data "exec 'Z99999.source(guessnum)'"
I'm thinking of a number between 1 and 10.
What is your guess?

Monorail:~ jbisti$ zowe tso send as Z99999-120-aabgaaam --data "1"
That's not it. Try again
What is your guess?

Monorail:~ jbisti$ zowe tso send as Z99999-120-aabgaaam --data "2"
That's not it. Try again
What is your guess?

Monorail:~ jbisti$ zowe tso send as Z99999-120-aabgaaam --data "3"
That's not it. Try again
What is your guess?

Monorail:~ jbisti$ zowe tso send as Z99999-120-aabgaaam --data "4"
You got it! And it only took you 4 tries!
READY
```

É possível ver que os comandos usam a Chave de Espaço de Endereço para assegurar que sua entrada continue indo para o espaço de endereço do TSO correto, que está lá esperando a próxima entrada.

Se você acertar na primeira tentativa, parabéns!

Sinta-se livre para iniciar o programa novamente e certifique-se de que você pode vê-lo através dos passos "Tente novamente".

Agora envie sua verificação de conclusão- **CHKREXX1** a partir de **ZXP.PUBLIC.JCL**

REXX1240416-1816

Bom trabalho-vamos recapitular	Em seguida ...
<p>Agora você está entrando no balanço das coisas. Você está interagindo com um programa Rexx, em execução em um espaço de endereço TSO, e você está fazendo isso através do comando zowe.</p> <p>Você provavelmente também aprendeu um pouco sobre a linguagem Rexx, e pode até ter feito alguma leitura extra sobre os Espaços de Endereço. Tudo aqui ajudará você a se tornar um profissional de mainframe mais qualificado.</p>	<p>O ferro é definitivamente fervente, e você provavelmente está se tornando um fã da Rexx. Por enquanto, no entanto, vamos continuar com o resto dos desafios em Fundamentos.</p>