Lição 3

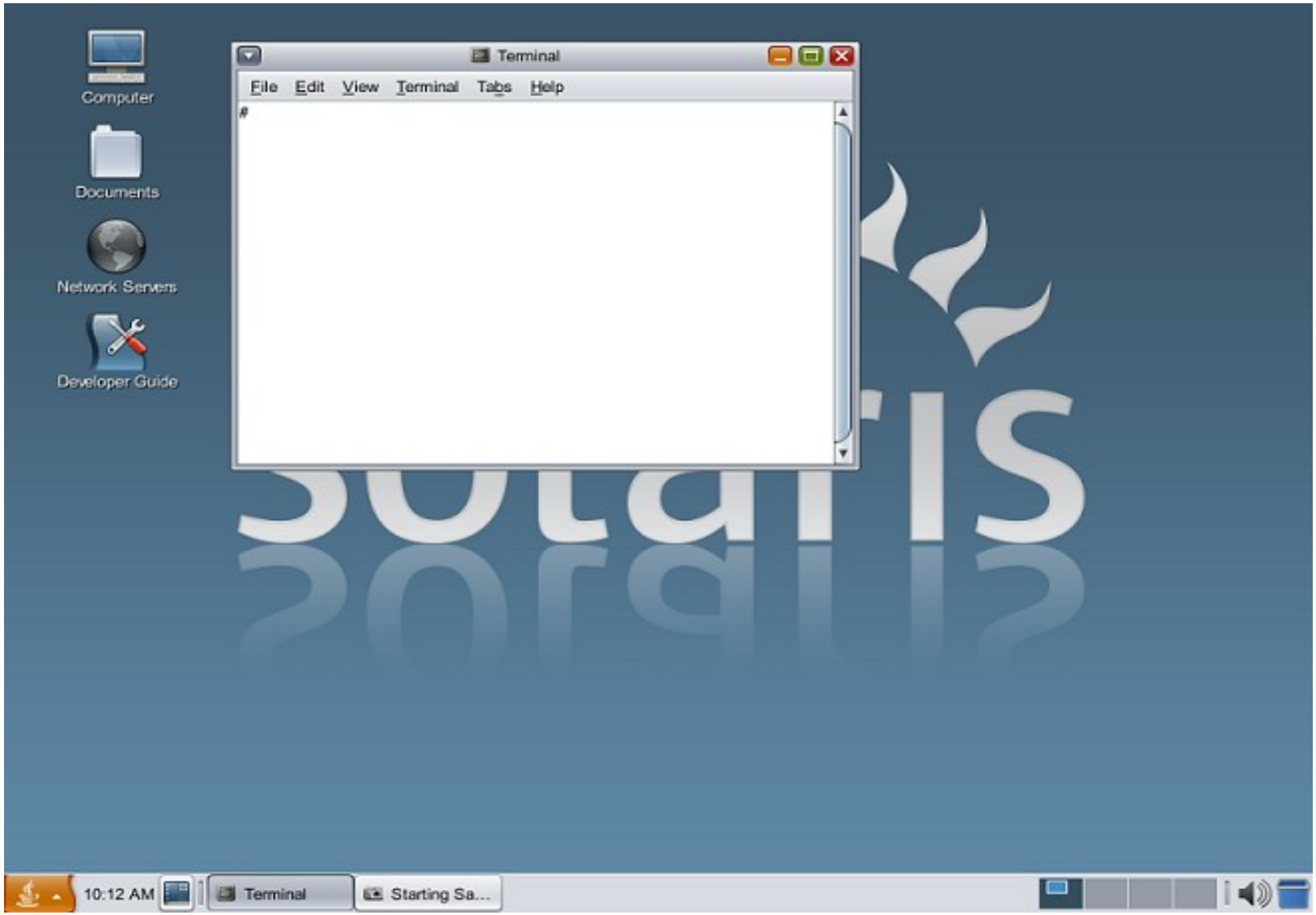
Comandos Básicos e Scripting

Objetivos

Ao final desta lição, o estudante será capaz de:

- Utilizar alguns comandos básicos em um Terminal
- Discutir a criação de *scripts* básicos e avançados
- Executar comandos básicos para administração do ambiente

Terminal


Computer
Documents
Network Servers
Developer Guide
Terminal
File Edit View Terminal Tabs Help
#
10:12 AM
Terminal
Starting Sa...

Comando de Ajuda

- Em cada terminal de comando está disponível a documentação de ajuda
- Acessar através do comando **man**, seguido pelo comando
- Para solicitar ajuda sobre o comando **ls**, digitar:

```
$ man ls
```

- A navegação pelo manual é feita utilizando as setas para cima e para baixo
- Para sair, pressionar a tecla **q**

Visualizar os Arquivos do Diretório Atual

- O comando **ls** lista os arquivos no diretório atual

```
$ ls /etc
```

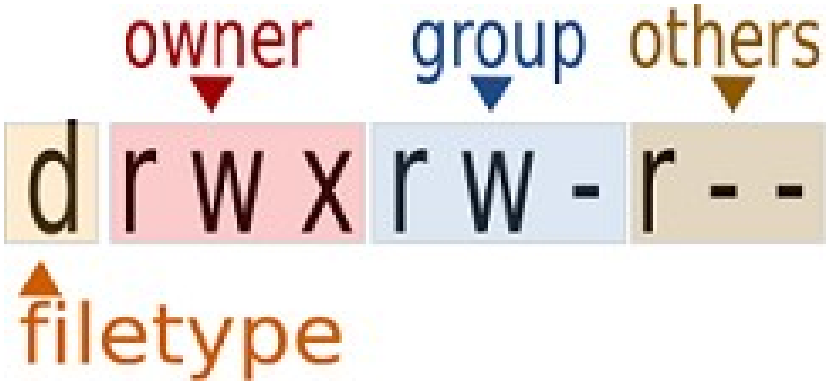
- Utilizado com curingas:
  - ***** representa 0 ou mais caracteres na sua posição
  - **?** representa um caractere qualquer na sua posição
- Pode ser executado com opções adicionais, especificadas por um sinal de menos:

-a (arquivos ocultos são incluídos)

-l (formato longo de exibição)

-sort (ordenar por um determinado operador)

Permissões de Arquivos




```
$ ls -l /etc
-rw-r--r--    1 root  root      753 Aug 20  2006 6to4.conf
drwxr-xr-x   13 root  root      442 Oct 19  2006 X11
-rw-r--r--    1 root  root      515 Aug 19  2006 afpovertcp.cfg
lrwxr-xr-x    1 root  root       15 Mar 23  2007 aliases -> postfix/aliases
-rw-r--r--    1 root  root    16384 Aug 20  2006 aliases.db
-rw-r--r--    1 root  root    17990 Mar 23  2007 authorization
-rw-r--r--    1 root  root      187 Aug 20  2006 bashrc
-rw-r--r--    1 root  root      137 Aug 20  2006 crontab
```

Obter a Localização Atual e Mudar de Diretório

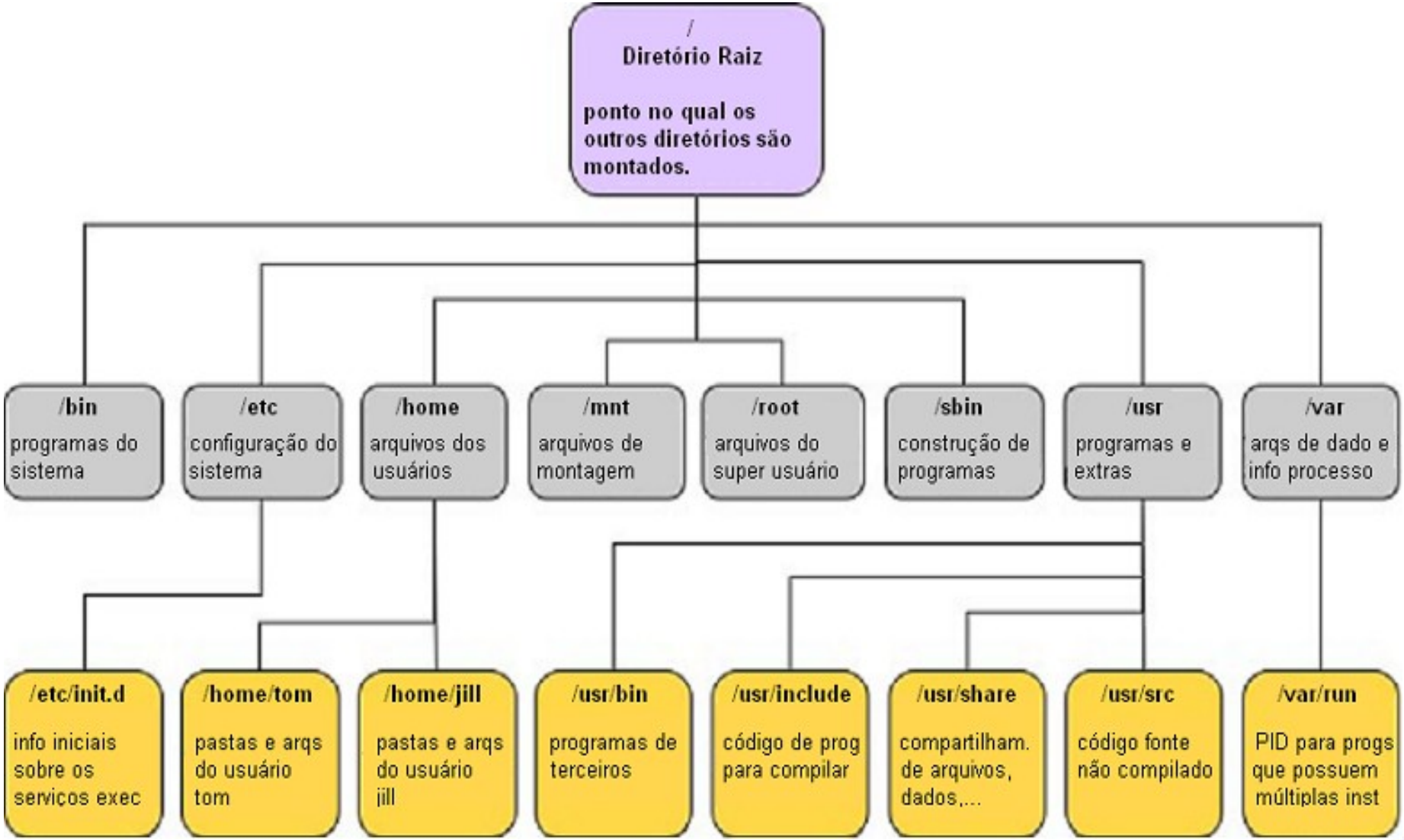
- Para saber o diretório atual, devemos executar o comando **pwd**

```
$ pwd
```

- Freqüentemente, seu diretório home é um diretório com seu nome de usuário no diretório **/export/home**
- Para mudar o diretório atual utilizamos o comando **cd**

```
$ cd <nome do diretório>
```

Sistema de Arquivos



Obter a Utilização e o Espaço Livre no Disco

- O comando **du** apresenta quanto espaço está sendo utilizado por arquivos em um diretório especificado

```
$ du -h /etc
```

- É frequentemente utilizado com o comando -h para facilitar sua leitura
- O comando **df** é utilizado para obter quanto espaço livre há em um disco

```
$ df -h
```

- Executando este comando no *prompt*, saberemos quanto espaço livre há disponível em todas as partições

Localizar Arquivos

- O comando **find** é utilizado para encontrar um determinado arquivo
- Pode ser especificada a opção **-name** para um determinado diretório
- É possível também procurar recursivamente
- Por exemplo, para visualizar todos os arquivos no diretório **/etc** que comecem com a palavra profile:

```
$ find /etc -name profile*
```

Copiar Arquivos

- O comando **cp** é utilizado para a copia de arquivos
- Possui como argumentos, o arquivo fonte e o de destino
- Estes arquivos podem conter abreviações, bem como curingas
- Por exemplo, para copiar o arquivo **passwd** do diretório **/etc** para seu diretório atual:

```
$ cp /etc/passwd
```

Mover Arquivos

- O comando **mv** é utilizado para mover arquivos
- Formato similar ao comando **cp**
- Ao término de uma cópia bem sucedida elimina os arquivos fontes
- Por exemplo, para mover o arquivo **passwd** do diretório **/etc** para seu diretório atual:

```
$ mv /etc/passwd
```

Eliminar Arquivos

- O comando **rm** é utilizado para eliminar arquivos
- Possui como argumento o arquivo a ser eliminado
- Este arquivo pode conter abreviações, bem como utilizar curingas
- Por exemplo, para remover o arquivo **passwd** do diretório **/etc**:

```
$ rm /etc/passwd
```

Criar e Eliminar Diretórios

- O comando **mkdir** é utilizado para criar diretórios
- Para criar um novo subdiretório **lesson1** em seu diretório atual:

```
$ mkdir lesson1
```

- O comando **rmdir** é utilizado para eliminar diretórios
- Para eliminar um subdiretório **lesson1** em seu diretório atual:

```
$ rmdir lesson1
```

Redirecionamentos

- A saída de um comando pode ser redirecionada para um arquivo
- Por exemplo, redirecionar a listagem do conteúdo do diretório **/etc** para um arquivo:

```
$ ls -l /etc > list.txt
```

- O operador **>** substitui o conteúdo anterior do arquivo destino
- O operador **>>** mantém o conteúdo anterior do arquivo destino

Paginar a Saída de Conteúdo

- O comando **more** é utilizado para visualizar o conteúdo na tela, uma página por vez

```
$ ls -l /etc | more
```

- O comando **less** é similar ao comando more, entretanto permite o retorno do conteúdo

```
$ ls -l /etc | less
```

Variáveis de Ambiente

- Variáveis de ambiente são variáveis definidas pelo operador do sistema
- Essa variáveis são identificadas pelo símbolo **$**
- Por exemplo, a variável $PATH lista os diretórios que o terminal procura para executar arquivos, quando o usuário executa um comando.
- Para saber o valor da variável PATH, você pode executar:

```
$ echo $PATH
```

Comando Echo e as Aspas

- Ao executar o comando **echo** você pode utilizar aspas simples ou duplas
- Uma variável colocada com aspas duplas é mostra seu conteúdo
  - A saída será o valor da variável e não seu nome;
  - Ao contrario do que acontece com a variável com aspas simples

```
$ echo 'Olá! Meu nome é $USER'
Olá! Meu nome é $USER
$ echo "Olá! Meu nome é $USER"
Olá! Meu nome é alice
```

Scripts

- Um arquivo que contém instruções sucessivas para o terminal
- Usaremos a linguagem shell **Bash**
- **Bash** é uma sigla para *Bourne-again shell*, que é uma revisão da linguagem *Bourne shell*
- Criar um script podemos utilizar qualquer editor

```
#!/bin/bash
# este é o meu primeiro roteiro bash.
echo 'Listando o conteúdo de /etc' > list.txt
ls -l /etc >> list.txt
echo 'Listando o conteúdo de /usr' >> list.txt
ls -l /usr >> list.txt
```

Execução do Script

- Descobrir quais são nossas permissões do roteiro do arquivo:

```
$ ls -l myscript
```

- Utilizar o comando **chmod** para tornar o arquivo executável:

```
$ chmod 755 myscript
```

- Para executar o script:

```
$ ./myscript
```

Comentários

- Comentários em um *script* **bash** é iniciado com o caracter #
  - Poderá haver um espaço entre o # e a primeira letra do seu comentário
- A primeira linha do roteiro bash não é realmente um comentário, mas um indicador em cada linguagem manuscrita para executar o roteiro
  - #!/bin/bash informa o SO para usar bash para executar o roteiro
  - #!/bin/ksh informa o SO para usar outro roteiro de linguagem (korn shell) para executar o roteiro

Construindo um script embutido

- Quando **bash** inicia primeiro em uma inicialização, ele executa comandos a partir do arquivo de roteiro **/etc/profile**.
  - O caminho do sistema será definido aqui
- Durante o **log-in do** usuário serão lidos e executados os arquivos ocultos .bash_profile , .bash_login e .profile do diretório home.
- Quando houver um login shell, Bash lerá e executará os comandos do arquivo. bash_logout

Scripts Avançados

- Os scripts também podem transmitir uma série de comandos
- Bash tem construtores para o comando de decisão e de repetição
- Podemos escrever scripts, que recebam informações do usuário

Substituindo Variáveis

- Uma variável é indicada com um sinal $
- Considere o comando abaixo:

```
$ x=42
$ echo $x
```

- A variável x contém o valor 42
- O comando **echo $x** é substituído por **echo 42**
- Uma nova linha pode ser impressa com o uso de barra invertida n

```
$ echo 'Olá \n mundo'
Olá \n mundo
$ echo 'Ola' $'\n' 'mundo'
Olá
Mundo
```

Variáveis Posicionais

- Variáveis especiais de $1 a $9 são utilizadas para substituírem os argumentos que podem ser passados aos *scripts*

```
#!/bin/bash
echo 'Meu primeiro argumento' $1
echo 'Meu segundo argumento' $2
echo 'Número de argumentos transmitidos' $#
```

- Para variáveis acima $ 9, o valor deve ser colocado entre chaves

```
echo 'Meu décimo argumento' ${10}
```

Comando Read

- Podemos receber um atributo através do comando read
- Por exemplo, o *script* abaixo solicita um nome:

```
#!/bin/bash
echo "Digite seu nome:"
read n
echo "Ola," $n "!"
```

Código de Erro

- Todos os comandos na maioria dos sistemas UNIX têm um código de erro que variam de 0 a 255
- Por padrão um programa, que teve sua execução bem sucedida retorna 0. Qualquer outro valor significa erro na execução
- Exemplo:

```
#!/bin/bash
ls $1
echo 'O código de erro do comando ls é: ' $?
```

Estruturas

- Os operadores aritméticos são os habituais +, -, ou *, /. O operador % retorna o resto da divisão de inteiros.
- O resultado de uma expressão aritmética pode ser atribuído a uma variável usando o comando let

```
$ x=5
$ let "x = $x + 1"
```

- O comando ((<expressão>)) avalia a expressão dentro do duplo parênteses

```
$ x=$((5 + 5))
```

Estrutura Condicional

- Uma expressão conditional Bash é verdadeira se seu retorno for igual a zero
- Para testar uma condição utilizamos uma condição **if**, como em uma linguagem de programação

```
if [condição] then
  <declaração>
elseif [condição] then
  <declaração>
else
  <declaração>
fi
```

Estrutura Condicional

- Podemos observar abaixo as comparações de uma condição **if**
- Note que "$a" e "$b" podem ser variáveis ou valores literais
  - if ["$a" -eq "$b"] - equivalência numérica
  - if ["$a" = "$b"] - equivalência entre expressões
  - if ["$a -ne "$b"] - diferença numérica
  - if ["$a" != "$b"] - diferença entre expressões
  - if ["$a" -gt "$b"] - Numericamente maior
  - -ge, -lt, -le – maior ou igual a, menor que, menor ou igual a
  - -n, -z – comparação not null ou null string
  - -f – se o nome do arquivo existir

Estrutura Condicional

- Expressões compostas significa combinar duas ou mais expressões conditionais
  - if [<condição1>] && [<condição2>] - operador **and**
  - if [<condição1>] || [<condição2>] - operador **or**
  - if [! <condição>] - sem operador

Estrutura de Repetição for

- Sintaxe:

```
for var in <list> do
  <declaração (cálculos)>
done
```

- **List** é a lista de que serão atribuidos à variável var

  No primeiro passo var assume o valor do primeiro elemento da lista, no segundo passo o do segundo e assim por diante

```
for days in "Seg" "Ter" "Qua" "Qui" "Sex" "Sáb"
  "Dom" do
   echo $days
done
```

Estrutura de Repetição While

- Testa uma condição e repete as declarações enquanto a condição for verdadeira

```
while [condição] do
  <declaração>
done
```

- Por exemplo:

```
#!/bin/bash
y=0
while [$y -l $1] do
  echo 'Olá Mundo!!'
  let "y = y + 1"
done
```

Mudando para Administrador

- A administração do sistema só pode ser realizada pelo super usuário ou usuário administrador (root)
- Para acessar a conta do administrador:

```
$ su
Digite sua senha: *******
#
```

- Também é possível usar o comando su para mudar o usuário

Criando Usuários

- Para acrescentar usuários ao Solaris, você pode utilizar o comando useradd
- Por exemplo, o comando a seguir acrescenta o usuário alice

```
# useradd -d /export/home/alice -m -s /bin/bash
  alice
```

- Vemos outras opções abaixo:
  - **-d** especifíca o diretório home para o usuário. Deve ser configurado em **/export/home**
  - **-m** informa **useradd** para criar manualmente o diretório
  - **-s** especifíca o terminal **shell**, que o usuário alice usará, neste caso, **bash**

Modificar a senha

- Para modificar a senha de um usuário, você pode executar o comando **passwd**
- Para modificar a senha do usuário alice digite:

```
# passwd alice
```

- Se nenhum parâmetro for fornecido, isso mudará a senha do usuário atual
- Este comando também pode ser utilizado por usuários comuns para alterar suas senhas

Remover usuários

- Para excluir usuários, você pode utilizar o comando **userdel**
- Por exemplo, para excluir o usuário alice:

```
# userdel -r alice
```

- **-r** especifica o diretório do usuário a ser removido

Sumário

- Terminal
- Comando de Ajuda
- Trabalhando com arquivos
- Variáveis de ambiente
- Scripts
- Parâmetros para Scripts
- Estrutura Condicional
- Estrutura de Repetição
- Administração do Sistema

Parceiros

- **Os seguintes parceiros tornaram JEDI™ possível em Língua Portuguesa:**