Parte 1: Conceitos Básicos de Terminal e Shell

1. O Que é o Shell?

O Shell é a camada mais externa que envolve o sistema operacional, funcionando como um intérprete de linha de comando. Ele permite que o usuário interaja com o sistema através de comandos digitados.

O Terminal é um ambiente de texto onde esses comandos são digitados e os resultados são exibidos. Ele é também chamado de Console quando em forma física, como teclados conectados diretamente a um sistema.

2. Pseudo-Terminais

Um pseudo-terminal é um emulador de terminal, permitindo simular um terminal físico em sistemas modernos.

Para abrir um pseudo-terminal, normalmente utiliza-se a combinação de teclas CTRL + Alt + T. Para fechar, basta digitar CTRL + D.

Cada pseudo-terminal ganha um arquivo de dispositivo no sistema, como mostrado pelo comando $ tty, que pode retornar algo como /dev/pts/0. Isso permite que o sistema identifique e interaja com diferentes terminais.

3. Operação Multiusuário

O sistema pode ser operado por múltiplos usuários, tanto localmente quanto remotamente:

Local: Usuários que operam diretamente na máquina.

Remoto: Usuários que operam via rede, utilizando protocolos como SSH ou Telnet.

Parte 2: Tipos de Shell e Comandos Básicos

1. Bourne Shell e BASH

O Bourne Shell (sh) foi desenvolvido por Stephen Bourne nos laboratórios da AT&T e foi o shell padrão do UNIX na versão 7, lançada em 1977.

O BASH (Bourne-Again SHell) foi criado pela Free Software Foundation (FSF) em 1989, como parte do projeto GNU. Ele é um shell gratuito e um dos mais usados até hoje. O BASH é uma evolução do Bourne Shell, oferecendo mais funcionalidades e sendo altamente utilizado para executar scripts.

2. ZShell (ZSH)

O ZSH é uma versão aprimorada do shell com funcionalidades adicionais em comparação ao BASH. Ele inclui recursos do KornShell (ksh) e do C shell (csh).

Desde 2019, o ZSH tornou-se o shell padrão no macOS, substituindo o BASH.

Uma popular ferramenta associada ao ZSH é o Oh My Zsh, que permite personalizar o shell com temas e plugins, facilitando a configuração e a customização.

3. Atalhos Comuns no Terminal

Existem diversos atalhos no terminal para facilitar o uso, como:

CTRL + Alt + T: Abrir o terminal.

CTRL + D: Fazer log off.

CTRL + C: Matar um processo.

CTRL + L: Limpar a tela.

CTRL + U: Limpar a linha antes do cursor.

CTRL + E: Ir ao fim da linha.

CTRL + Y: Colar o texto.

CTRL + R: Buscar um comando recente.

4. Usuário Root vs. Usuário Comum

No terminal, o prompt de comando termina com um caractere especial que indica o tipo de usuário:

#: Indica que o usuário é o root (administrador do sistema).

$: Indica que o usuário é um usuário comum.

O comando whoami pode ser usado para identificar o usuário ativo no terminal.

Parte 3: Estrutura de Diretórios no Sistema Operacional

1. Estrutura de Diretórios

A estrutura de diretórios de um sistema UNIX/Linux segue uma hierarquia organizada, começando pela raiz representada por /. Abaixo dela, encontram-se diversos diretórios que servem para diferentes propósitos:

/bin: Contém arquivos essenciais, como programas e comandos necessários para a recuperação do sistema.

/boot: Armazena os arquivos necessários para inicializar o sistema, incluindo o Kernel e o bootloader GRUB.

/dev: Contém arquivos de dispositivos, que permitem a interação do sistema operacional com o hardware (por exemplo, discos e portas USB).

/etc: Contém arquivos de configuração do sistema e da rede.

/home: Diretório que armazena os arquivos e configurações pessoais de cada usuário.

/lib: Contém bibliotecas e módulos necessários para o funcionamento de programas armazenados em /bin e /sbin.

/media: Ponto de montagem automático para mídias removíveis como CD-ROMs e pendrives.

/mnt: Ponto de montagem manual usado por administradores de sistema para acessar sistemas de arquivos.

/opt: Contém programas de terceiros que não fazem parte da distribuição oficial do sistema (ex.: Google Chrome).

/proc: Sistema de arquivos virtual que contém informações sobre processos e o estado atual do sistema.

/tmp: Armazena arquivos temporários que geralmente são excluídos após o reinício do sistema.

/usr: Contém a maior parte dos programas e arquivos compartilhados entre os usuários, como binários em /usr/bin e bibliotecas em /usr/lib.

/var: Armazena arquivos que variam constantemente, como logs, filas de spool e bancos de dados.

2. Comandos para Navegação entre Diretórios

Para navegar entre diretórios no terminal, utilizam-se os seguintes comandos:

cd [diretório]: Entra no diretório especificado.

cd ..: Sobe um nível na hierarquia de diretórios.

cd /: Vai para o diretório raiz.

cd ~: Vai para o diretório home do usuário atual.

cd -: Retorna ao diretório anterior.

3. Comando su (Substitute User)

O comando su permite alternar entre diferentes usuários no terminal:

su [usuário]: Alterna para o usuário especificado.

su -: Alterna para o usuário root, carregando o ambiente completo.

su -c "[comando]" [usuário]: Executa um comando com os privilégios de outro usuário sem trocar de sessão.

Parte 4: Edição de Texto no Shell

1. Editores de Texto no Shell

No ambiente Shell, há vários editores de texto que permitem a edição de arquivos diretamente no terminal. Dois dos mais comuns são o vi e o nano, que vêm instalados na maioria das distribuições Linux.

vi é um editor mais avançado e poderoso, ideal para programadores e usuários experientes. Ele possui dois modos principais: modo de comando e modo de inserção. Já o nano é um editor mais simples e intuitivo, adequado para tarefas rápidas de edição de texto.

2. Emacs

Emacs é outro editor de texto amplamente utilizado, desenvolvido por Richard Stallman em 1976. Ele é altamente configurável e permite que programadores escrevam e modifiquem código de maneira eficiente.

No Emacs, as combinações de teclas são muito usadas para realizar ações, como:

C-x C-f: Abrir um arquivo.

C-x C-s: Salvar o arquivo atual.

C-x C-c: Fechar o editor.

C-f: Avançar um caractere.

C-b: Voltar um caractere.

C-n: Descer uma linha.

C-p: Subir uma linha.

Além disso, o Emacs permite split windows (divisão da tela) para trabalhar em vários arquivos simultaneamente, com comandos como:

C-x 2: Dividir a janela horizontalmente.

C-x 3: Dividir a janela verticalmente.

C-x o: Alternar entre as janelas.

3. Atalhos no Emacs

No Emacs, muitos atalhos facilitam a navegação e edição de texto:

M-w: Copiar.

C-y: Colar.

C-w: Recortar.

C-_: Desfazer.

C-x h: Selecionar todo o conteúdo.

C-s: Buscar para frente.

C-r: Buscar para trás.

4. Comandos Estendidos no Emacs

Emacs também possui uma função chamada M-x que permite executar comandos estendidos, como:

M-x save-buffer: Salvar o buffer atual.

M-x search-forward: Buscar uma string à frente.

M-x shell: Executar um terminal dentro do próprio Emacs.