

LXT0203-Uso básico do terminal Linux: Parte 1 – Shell

📅 Data do post	@June 29, 2022 9:00 AM (GMT)
🔑 Palavras-chave	Uso básico da Linha de Comandos
☑ Publicado	<input type="checkbox"/>
📌 Pronto	Preparado
☑ IG Post	<input checked="" type="checkbox"/>
🌟 Status	Done

Linkedin-Post

A linha de comandos (terminal) é um front-end ou ponte de comunicação para o Shell.

Todo comando digitado no terminal Linux é enviado para o Shell que, por sua vez, interpreta e/ou processa a acção e o resultado é devolvido ao terminal Linux uma saída do comando digitado.

Esse artigo fala sobre conceitos básicos e do Shell e alguns comandos que permitem ter informações sobre o Shell. Leia o artigo na íntegra.

Siga-me e veja outros posts no meu Instagram <https://www.instagram.com/idalectiosilvatech/>

Deixe o like para medirmos a repercussão do conteúdo.

Guarde para rever noutro momento.

Compartilha o conteúdo com os amigos e não só

Comente o que acha do conteúdo e da iniciativa e deixe sugestões.

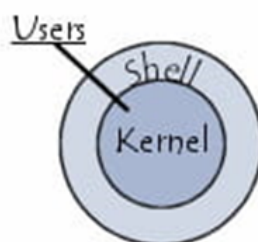
[#cybersecurity](#) [#technology](#) [#linux](#) [#CLI](#) [#Shell](#)

Introdução

Os computadores possuem diversas características. Mas, a que mais destaca-se é a interacção humano-computador. Essa interacção pode ser feita de várias maneiras, podendo ser através de um teclado, mouse e monitor. A interface que permite essa interacção é conhecida como **interface gráfica do usuário** (GUI - *graphical user interface*).

Nos primeiros anos da computação, a interacção era diferente, pois era baseada em comandos digitados em uma interface de linha de comandos (CLI, *command-line interface*) ou terminal.

Sendo que os comandos são digitados e depois enviados para o computador ler e processar e depois executar uma saída, é necessário um programa intermediário que funciona como tradutor. Por esse motivo, foi desenvolvido o Shell.



Características da Linha de Comandos

Por padrão, na maior parte dos Shells, o prompt é composto pelo nome da máquina, seguido de dois-pontos (:), do directório actual e de um caractere que indica o tipo de usuário conectado: `$` indica que se trata de um usuário normal e `#` do administrador, chamado root.

A maioria dos comandos na linha de comando segue a mesma estrutura básica:

```
comando [opção(ões)/parâmetro(s)...] [argumento(s)...]
```

Comando: Programa que o usuário pretende executar.

Exemplo: `$ ls -l /home`

Opção(ões)/Parâmetro(s): um “botão” que modifica o comportamento do comando de alguma forma e podem ser acessadas na forma curta ou longa.

Curta: `ls -l /home`

Longa: `ls --format = long /home`

Argumento(s): dados adicionais exigidos pelo programa, como um nome de arquivo ou caminho, como /home.

Curta: `ls -l /home`

A única parte obrigatória dessa estrutura é o próprio comando. Em geral, todos os outros elementos são opcionais, mas um programa pode exigir que determinadas opções, parâmetros ou argumentos sejam especificados.

Sintaxe de comandos

Ao usar um Shell interactivo, o usuário insere comandos no chamado prompt. Para cada distribuição Linux, o prompt padrão tem uma aparência um pouco diferente, mas ele geralmente segue esta estrutura:

```
username@hostname directório_atual tipo_de_shell
```

Aonde:

- `Username` : Nome do usuário que executado no Shell
- `Hostname` : Nome da máquina hospedeira onde o Shell é executado. Também existe um comando `hostname`, que permite exibir ou definir o nome de hospedeiro do sistema.
- `directório_atual` : O directório em que o Shell está actualmente. Um til (~) indica que o Shell está no directório `home` do usuário actual.
- `Tipo_de_shell` : `$` indica que o Shell está sendo executado por um usuário comum.
`#` indica que o Shell está sendo executado pelo super usuário `root`.

Como não precisamos de privilégios especiais, usaremos um prompt sem privilégios nos exemplos a seguir. Por questões de concisão, usaremos apenas o `$` como prompt.

Nas distribuições baseadas no Debian GNU/Linux, o prompt de um usuário comum aparece da seguinte forma:

```
idaseg@debian:~$
```

O prompt do super-usuário terá a seguinte aparência:

```
root@debian:~#
```

No CentOS ou Red Hat Linux, o prompt para um usuário comum será como este: `[idaleciosilva@centos ~]$`

E o prompt do super-usuário será assim: `[root@centos ~]#`

Tipos de Shell

O Shell é um arquivo executável cuja função é ler os comandos digitados na linha de comandos, interpretar o seu significado e, devolver o resultado pelas saídas.

Existem vários tipos de Shell, os mais comuns:

sh (Bourne Shell)

O Bourne shell é chamada interface de linha de comando ou interpretador de comando, e é uma linguagem de script. É também conhecido como “sh”.

Como linguagem de script, o usuário pode usá-lo para estabelecer variáveis para capturar dados conhecidos ou desconhecidos de entrada, ou saída e manipular o processamento desses dados por meio do uso de declarações condicionais no script através de uma técnica conhecida como fluxo de controle.

bash (Bourne again shell)

O Bash é um interpretador de comandos que foi criado ou idealizado por Steven Bourne, em 1977, o Bash e passou a ser o acrónimo de **Bourne Again Shell**.

O Shell mais popular se chama **Bash**, e é o padrão na maioria das distribuições Linux, e falarei sobre ele no próximo artigo, não perca!

csh (C Shell)

O Shell C (csh) é um Shell de comando para sistemas do tipo Unix que introduziu várias melhorias em relação ao Shell Bourne anterior, projectado para uso interactivo.

O Shell C recebe o nome de sua sintaxe, que se parece com a linguagem de programação C, pois introduziu recursos destinados a facilitar o uso interactivo na linha de comando. Um dos recursos mais notáveis foi o histórico de comandos. Os usuários podem recuperar comandos anteriores inseridos e repeti-los ou editar esses comandos. Os aliases permitem que os usuários definam nomes abreviados para serem expandidos em comandos mais longos. Uma pilha de directórios permite que os usuários pressionem e poupem os directórios da pilha para ir e voltar rapidamente. O Shell C também introduziu a notação til padrão, em que "~" representa o directório inicial de um usuário.

Tcsh (Tenex C shell)

O **tcsh** é uma versão aprimorada, mas completamente compatível, do Shell Berkeley UNIX C, **csh**. É um intérprete de linguagem de comando utilizável como um Shell interactivo e um processador de comandos de script de Shell. Inclui um editor de linha de comando, conclusão programável de palavras, correcção ortográfica, um mecanismo de histórico, controle de trabalho e uma sintaxe semelhante à linguagem de programação C.

zsh (Zero shell).

O zsh é um interpretador de comando para shell scripting cuja finalidade é facilitar o uso da ferramenta. Entre suas características, destacam-se a gama de personalizações que podem ser aplicadas em sua interface e, principalmente, as optimizações disponíveis para o terminal do Linux. NOTA: normalmente, seus nomes correspondem ao nome do executável.

Comandos e arquivos de configuração no Shell Linux

Para saber qual é o Shell padrão do usuário, execute o comando

```
echo $SHELL
```

```
root@debian:~# echo $SHELL
/bin/bash
```

Para alterar Shell deve-se o arquivo de configuração correspondente que é o `/bin/bash`

Ver todos os usuários do sistema

Sintaxe do comando: `$ cat/etc/passwd`

```
idaseg@debian:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:109::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:110:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
idaseg:x:1000:1000:idaseg,,,:/home/idaseg:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
idaseg@debian:~$
```

Nesse arquivo também definimos o Shell que queremos usar no sistema, sempre o terminal for aberto.

Por outro lado, sendo que existem diversos tipos de Shell, podemos definir o Shell padrão é no arquivo de configuração

`/etc/passwd`

Mudar o Shell

Sintaxe do comando: `$ sudo usermod -s /usr/bin/zsh + nome-do-usuário`

```
idaseg@debian:~$ sudo usermod -s /usr/bin/zsh idaseg
[sudo] senha para idaseg:
idaseg@debian:~$ cat /etc/passwd | grep idaseg
idaseg:x:1000:1000:idaseg,,,:/home/idaseg:/usr/bin/zsh
```

Esse comando vai mudar o Shell de Bash para Zsh



OBS: temos de mudar o `/bin/bash`

Ver o Shell que está a ser usado

Sintaxe do comando01: `$ cat /etc/passwd | grep + nome-do-usuário`

```
idaseg@debian:~$ cat /etc/passwd | grep idaseg
idaseg:x:1000:1000:idaseg,,,:/home/idaseg:/bin/bash
```

Sintaxe do comando02: `$ which + nome-do-shell-à-procurar`

```
idaseg@debian:~$ which bash
/usr/bin/bash
```

Comando Essenciais do Shell

Comando `whoami`

Sintaxe do comando: `$ whoami`

```
idaseg@debian:~$ whoami
idaseg
idaseg@debian:~$ _
```

Quando digitamos `whoami` o shell:

1. Encontra um programa chamado `whoami`
2. Executa o programa
3. Mostra a saída do programa
4. Mostra um novo prompt para mostrar que está pronto para receber mais comandos

Comando `pwd`

Sintaxe do comando: `$ pwd`

```
idaseg@debian:~$ pwd
/home/idaseg
```

O comando `pwd` (*print working directory*) pode significar “diretório de trabalho” ou “diretório atual” ou “diretório padrão” permite descobrir aonde estamos - qual diretório do sistema de arquivos.

Comando `ls`

Sintaxe do comando: `$ ls`

```
idaseg@debian:~$ ls
Desktop  Documentos  Imagens  Músicas  Transferências
```

Esse comando mostra os nomes dos arquivos e diretórios no diretório atual em ordem alfabética. Como mostra a imagem, o comando `ls` listou ou mostra todo o conteúdo do diretório do usuário `idaseg`.

É possível também “listar” mais informações quando usamos esse comando quando usamos um **argumento adicional ao comando**. Esse é um dos chamados de comandos navegação, e destes o próximo artigo sobre Linux, vai mostrar muitos argumentos e opções que permitem alterar a saída do resultado.

Conclusão

Conhecer o Shell permite ter uma noção clara de qual o deve o Shell a usar de acordo com as funções que desempenha. Todos esses comandos até apresentados nesse artigo são os mais básicos que existem principalmente quando alguém pretende trabalhar com sistemas Linux quer seja analista ou administrador.

Gostou do artigo? Siga-me e veja outros posts no meu Instagram <https://www.instagram.com/idaeciosilvatech/>

Deixe o like para medirmos a repercussão do conteúdo.

Guarde para rever noutro momento.

Compartilha o conteúdo com os amigos e não só

Comente o que acha do conteúdo e da iniciativa e deixe sugestões.

[#cybersecurity](#). [#technology](#) [#linux](#) [#CLI](#) [#Shell](#)