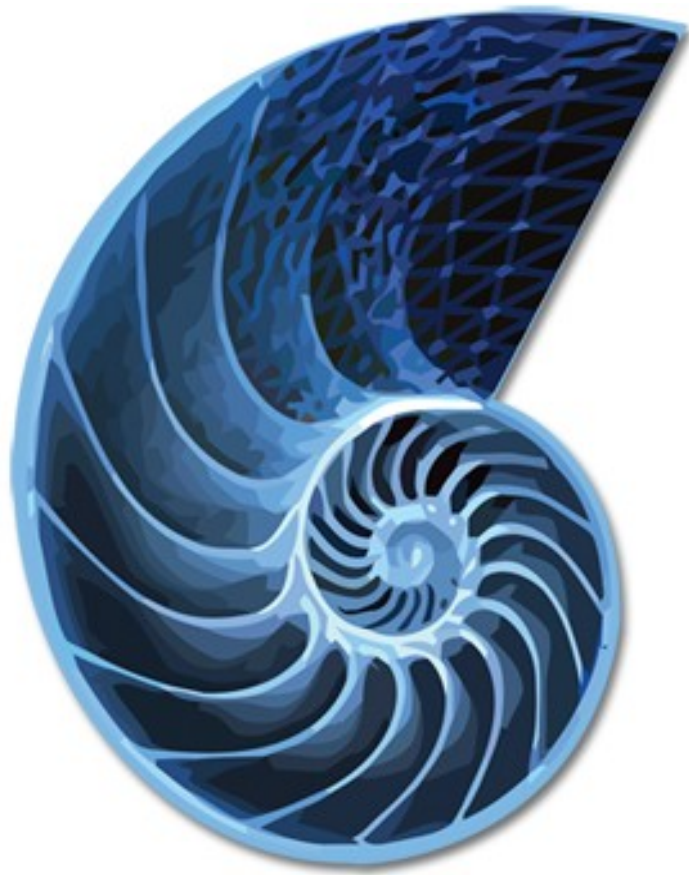
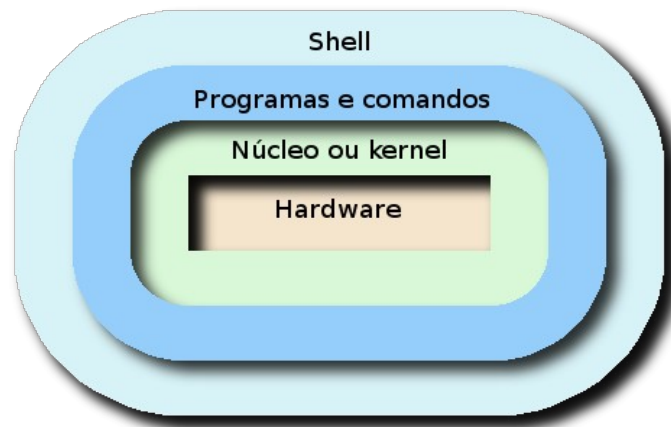


Shell Script



Eduardo Mondoni
mondoni@linustec.com.br

Camadas do Linux



Com base na figura acima, podemos compreender quais são as camadas que fazem parte de um sistema operacional GNU/Linux desde o Hardware até o Shell, que é o objetivo desta apostila.

- Hardware: composto pelos dispositivos físicos do sistema (teclado, mouse, monitor).
- Kernel: núcleo do sistema operacional sendo responsável pelo controle e gerenciamento do hardware.
- Programas: comandos que realizam tarefas bem específicas.
- Shell: responsável pela interação entre o usuário e o sistema operacional.

O que é o Shell?

Shell é o programa que reconhece o comando que você digitou e traduz para o sistema operacional qual a ação que será realizada, diminuindo o tempo de utilização do kernel para a execução da ação.

Por ser um interpretador de comandos de alto nível, o shell também pode ser utilizado como linguagem de programação. Onde o usuário pode combinar uma sequência de comandos criando um script para ser executar ao invés de sempre realizar a mesma sequência de comandos manualmente.

Tipos de Shells

- **Bourne Shell (sh):** Desenvolvido por Stephen Bourne da Bell Labs (da AT&T onde também foi desenvolvido o Unix), este foi durante muitos anos o Shell default do sistema operacional Unix. É também chamado de Standard Shell por ter sido durante vários anos o único e até hoje é o mais utilizado até porque ele foi portado para todos os ambientes Unix e distros Linux.
- **Korn Shell (ksh):** Desenvolvido por David Korn, também da Bell Labs, é um superset do sh, isto é, possui todas as facilidades do sh e a elas agregou muitas outras. A compatibilidade total com o sh vem trazendo muitos usuários e programadores de Shell para este ambiente.
- **Bourne Again Shell (bash):** Este é o Shell mais moderno e cujo número de adeptos mais cresce em

todo o mundo, seja por ser o Shell default do Linux, seu sistema operacional hospedeiro, seja por sua grande diversidade de comandos, que incorpora inclusive diversas instruções características do C Shell.

- **C Shell (csh):** Desenvolvido por Bill Joy da Berkley University é o Shell mais utilizado em ambientes *BSD e Xenix. A estruturação de seus comandos é bem similar à da linguagem C. Seu grande pecado foi ignorar a compatibilidade com o sh, partindo por um caminho próprio. Além destes Shells existem outros, mas irei falar contigo somente sobre os três primeiros, tratando-os genericamente por Shell e assinalando as especificidades de cada um que porventura hajam.

Vantagens do Bourne Again Shell

Bash é um shell compatível com sh que incorpora funções úteis de Korn shell (ksh) e C shell (csh). Está em conformidade com o padrão IEEE POSIX P1003.2/ISO 9945.2 Shell and Tools. Oferece melhorias em relação ao sh tanto para programação quanto utilização interativa. Incluindo, edição de linha de comando, histórico de comandos ilimitado, controle de jobs entre outros. O bash pode executar a maioria dos scripts sh sem modificações.

Arquivos de inicialização do bash

Arquivos de inicialização são script que serão lidos e executados pelo Bash ao ser iniciado. A seguir uma breve explicação das diferentes maneiras de iniciar o bash e que arquivos serão lidos na inicialização.

Shell Interativo com login ou opção “--login”

Significa que você está logado num shell após ter efetuado login com usuário e senha, não necessariamente existirá um script sendo executado.

Arquivos lidos:

- /etc/profile
- ~/.bash_profile, ~/.bash_login ou ~/.profile: será lido o primeiro arquivo existente.
- ~/.bash_logout: lido no momento de efetuar logout.

Shell Interativo sem login

Significa que você não precisou se autenticar no sistema. Exemplo, quando você abre um terminal a partir de um ícone ou um atalho de menu, isso é um shell sem login.

Arquivos lidos:

- ~/.bashrc

Este arquivo é geralmente citado em ~/.bash_profile:

```
if [ -f ~/.bashrc ]; then . ~/.bashrc; fi
```

Shell não interativo

Todos os script utilizando shell não interativo. Eles são programados para realizarem certas tarefas e não podem ser instruídos a realizarem outros trabalhos diferentes dos programados.

Arquivos lidos:

- definido por BASH_ENV.

Execução com o comando sh

O bash atua como o programa Bourne sh.

Arquivos lidos:

- /etc/profile
- ~/.profile

Executando comandos

O bash determina o tipo de programa que será executado. Programas normais são comandos do sistema que existem na forma compilada. Quando um desses programas é executado, um novo processo é criado porque o Bash faz uma cópia exata dele mesmo. Este procedimento chama-se *forking*.

Comandos built-in do Shell

São comandos existentes dentro do próprio shell. Quando um desses comandos é executado sendo a primeira palavra de uma linha de comando, o shell o executa diretamente sem a necessidade de criar um novo processo.

Propriedades de bons scripts:

1. Um script deve executar sem erros;
2. Deve executar a tarefa para qual foi criado;
3. Lógica de programação é claramente definida;
4. Um script não realiza tarefas desnecessárias;
5. Scripts devem ser reutilizáveis;

Nosso primeiro script

Como exemplo do primeiro script que iremos utilizar, iremos criar um arquivo chamado *helloworld.sh* com o seguinte conteúdo:

```
#!/bin/bash  
echo "Hello World"
```

Antes de executar o script, vamos configurar as permissões de execução:

```
$ chmod + x helloworld.sh
```

E agora iremos executar:

```
$ ./helloworld.sh
```

Como resultado temos:

```
$ ./helloworld.sh  
Hello World
```

Exemplos

Todos os exemplos utilizados nessa apostila pode ser encontrados no seguinte link hospedado no github:

<https://github.com/mondoni/linuxcourse/tree/master/shellscript>

Referências

Neves, Julio Cezar. Programação Shell Linux, sétima edição, 2008.

Garrels, machtelt. Bash Beginners Guide, versão 1.1, 2008.

<http://marcosabadi.blogspot.com/2011/12/uma-rapidinha-nos-principais-sabores-de.html>