

Certificação LPIC 1

```
[root@dev ~]# cat hello.sh
```

```
#!/bin/bash  
echo "Bem vindos!";
```

```
[root@dev ~]# whoami  
Rafael Rêgo
```

```
[root@dev ~]# mail -s 'Contato'  
rafael.force@gmail.com, whatsapp 98860-7630
```

```
[root@dev ~]# cat today.txt
```

- **Tópico 101: Arquitetura do sistema**
- **101.1 Determinar e configurar o hardware**
 - **O candidato deve estar apto a configurar dispositivos de hardware do sistema**

```
[root@dev ~]# cat intro.txt
```

- O Linux suporta uma grande quantidade de dispositivos e basicamente faz isso via módulos
- Os dispositivos disponíveis para o Linux precisam ter módulos que podem ser carregados na inicialização ou com o sistema já inicializado
- Ou podem ser descarregados interrompendo a funcionalidade dos dispositivos
- O sistema udev nomeia dinamicamente e cria os dispositivos lógicos para acesso a esse hardware detectado

```
[root@dev ~]# cat intro.txt
```

- E o HAL (hardware abstraction layer) passa informações sobre mudanças de hardware pra camada do usuário (user space), mas o HAL está sendo incorporado pelo udev
- O udev é o sucessor do devfs e do hotplug e está incorporando o HAL e sua função básica é gerenciar dispositivos do sistema, nomeando dinamicamente e localizando abaixo do /dev

```
[root@dev ~]# cat filesystem.txt
```

→ Os principais diretórios no sistema operacional que criam apontamentos para dispositivos do sistema são os seguintes:

- **/sys**
- **/proc**
- **/dev**

[root@dev ~]# apropos sysfs

- Introduzido no kernel 2.6 o sysfs é uma união do proc, devfs e devpty
- O sysfs enumera os dispositivos e barramentos conectados ao sistema em uma hierarquia no sistema de arquivos que pode ser acessado pelo usuário
- O sysfs é montado no /sys e contém diretórios que organizam os dispositivos conectados de várias formas diferentes

[root@dev ~]# apropos sysfs

→ A estrutura de diretórios do sysfs contém:

- /devices/: Representa através de subdiretórios todos os subcanais detectados pelo kernel (endereço lógico para acesso ao dispositivo)**
- /bus/: Contém dispositivos acessados via canal e que podem usar mais de um subcanal por dispositivo**
- /class/: Contém diretórios que agrupam dispositivos similares como ttys, SCSI tapes, dispositivos de rede etc.**

[root@dev ~]# apropos sysfs

→ A estrutura de diretórios do sysfs contém:

- /block/: Representa através de subdiretórios todos os subcanais detectados pelo kernel (endereço lógico para acesso ao dispositivo)**

Certificação LPIC 1

[root@dev ~]# apropos sysfs

- Os principais comandos para trabalhar com hardware no linux são os seguintes:
- lsusb, lspci, modprobe, lsmod, uname, lshw, lscpu, lsblk, lsscsi, dmidecode

[root@dev ~]# lsusb

- **Instalação: yum install usbutils**
- **Descrição: apresentar dispositivos USB e barramento se existentes**
- **Principais parâmetros:**
 - **-v: apresenta informações detalhadas sobre cada dispositivo USB**
- **Manpage: man lsusb**

[root@dev ~]# lspci

- **Instalação: yum install pciutils**
- **Descrição: apresentar dispositivos e barramentos PCI**
- **Principais parâmetros:**
 - **-v: apresenta informações detalhadas sobre cada dispositivo PCI**
 - **-t: apresenta a saída em forma de árvore para facilitar o entendimento da hierarquia**
- **Manpage: man lspci**

[root@dev ~]# modprobe ip-tables

- **Instalação: [base]**
- **Descrição: carregar ou remover módulos**
- **Principais parâmetros:**
 - **-f: força ou inserção ou remoção do módulo**
 - **-D: imprime as dependência do módulo**
 - **-r: remove o módulo ou invés de inserir**
- **Manpage: man modprobe**

Certificação LPIC 1

[root@dev ~]# lsmod

- **Instalação: [base]**
- **Descrição: listar módulos carregados**
- **Manpage: man lsmod**

[root@dev ~]# uname -a

- **Instalação: [base]**
- **Descrição: mostra informações da arquitetura do hardware e do kernel**
- **Principais parâmetros:**
 - **-a: lista todas as informações disponíveis**
- **Manpage: man uname**

```
[root@dev ~]# lshw -c net
```

- **Instalação: yum install lshw**
- **Descrição: apresenta informações detalhadas da configuração de hardware da máquina**
- **Principais parâmetros:**
 - **-c: apresenta informações apenas da classe informada**
- **Manpage: man lshw**

[root@dev ~]# lscpu

- **Instalação: [base]**
- **Descrição: apresenta informações da arquitetura da cpu a partir do /proc e /sys**
- **Principais parâmetros:**
 - **-a: apresenta informações apenas de processadores online e offline**
- **Manpage: man lscpu**


```
[root@dev ~]# lsblk
```

- Instalação: [base]**
- Descrição: lista informação de todos os dispositivos de bloco**
- Nota:**
 - Major: seleciona qual device driver é chamado para operação de I/O**
 - Minor: é passado como parâmetro para chamada da operação de I/O**
- Manpage: man lsblk**

```
[root@dev ~]# lsscsi -g
```

- **Instalação: [base]**
- **Descrição: lista informações de dispositivos sas, sata e scsi conectados ao sistema, consulta informações do sysfs**
- **Principais parâmetros:**
 - **-g: mostra informações dos drives genéricos**
 - **-c: saída classica do comando**
- **Manpage: man lsscsi**

[root@dev ~]# dmidecode

- **Instalação: [base]**
- **Descrição: apresenta uma descrição dos componentes de hardware do sistema**
- **Manpage: man dmidecode**

```
[root@dev ~]# cat today.txt
```

- **Tópico 101: Arquitetura do sistema**
- **101.2 Boot do sistema**
 - **O candidato deve estar apto a guiar através do processo de boot do sistema**

Certificação LPIC 1

```
[root@dev ~]# cat intro.txt
```

- O gerenciador de boot padrão do linux é o Grub que atualmente está na versão 2
- SYSLINUX também é um outro gerenciador de boot do linux, mas é mais usado para boot de Cds e Dvds e seu uso para boot de sistemas linux é desaconselhado
- O principal arquivo de configuração do grub2 é /boot/grub2/grub.cfg
- Diferente da versão 1 do grub a versão 2 é manipulada a partir do arquivo /etc/default/grub e do diretório /etc/grub.d

```
[root@dev ~]# cat intro.txt
```

- O arquivo **grub.cfg** contém sintaxe de script incluindo comandos de loop, funções e variáveis
- Os dispositivos de disco agora são identificados por **UUID** (Indetificador Único Universal) para evitar perda de boot em caso de troca do nome do dispositivo (Ex: **/dev/sda** passar a ser **/dev/sdb**)
- Vamos analisar alguns arquivos em **/etc/grub.d/ ...**

[root@dev ~]# apropos grub2

- Os principais comandos para trabalhar com o grub são os seguintes:
 - grubby, grub2-mkconfig, grub2-set-default

[root@dev ~]# grubby

- **Instalação: [base]**
- **Descrição: linha de comando para configurar o grub**
- **Exemplos:**
 - **grubby --default-kernel**
 - **grubby --default-index**
 - **grubby -info=ALL**
 - **grubby --set-default /boot/vmlinuz-x.y.z-a.b.c**
- **Manpage: man grubby**


```
[root@dev ~]# grub2-mkconfig
```

- **Instalação: [base]**
- **Descrição: Gera um arquivo de configuração do grub2**
- **Exemplos:**
 - **grub2-mkconfig -o /boot/grub2/grub.cfg**
- **Manpage: man grub2-mkconfig**

[root@dev ~]# grub2-set-default

- **Instalação: [base]**
- **Descrição: Define a opção default de boot pro grub**
- **Exemplos:**
 - **grub2-set-default 2**
- **Manpage: man grub2-set-default**

Certificação LPIC 1

```
[root@dev ~]# cat intro.txt
```

- O Linux tem dois métodos para gerenciar a inicialização de serviços e do próprio sistema que são:
 - System V initialisation
 - Systemd

Certificação LPIC 1

```
[root@dev ~]# cat intro.txt
```

→ A inicialização do Linux no modelo System V (SysVInit) é organizado na seguinte estrutura de diretórios:

➤ /etc/rc.d

init.d

rc0.d

rc1.d

rc2.d

rc3.d

rc4.d

rc5.d

rc6.d

```
[root@dev ~]# cat intro.txt
```

- Cada um desses índices (rc?.d) estava relacionado a um respectivo runlevel
- Estes são os seguintes runlevels possíveis:
 - 0 – hatl
 - 1 – modo single (mono usuário sem rede)
 - 2 – multiusuário sem rede (definido pelo usuário)
 - 3 – modo multiusuário com rede
 - 4 – não utilizado ou definido pelo usuário

Certificação LPIC 1

```
[root@dev ~]# cat intro.txt
```

- **5 – multusuário modo gráfico**
- **6 – reboot**

```
[root@dev ~]# cat intro.txt
```

- A seleção de qual runlevel será iniciado por padrão é definida no arquivo `/etc/inittab` e tem a seguinte sintaxe:
 - `id:5:initdefault`
- E o controle do que é iniciado ou não pelo sistema é feito por meio da ferramenta `chkconfig` ou `ntsysv` (ou a interface gráfica `system-config-services`)

```
[root@dev ~]# cat intro.txt
```

- O modelo systemd é mais centralizador e por meio de algumas ferramentas permite controlar melhor o sistema com menos contato com arquivos de configuração
- No próximo slide vamos apresentar algumas tabelas com comparativos entre os dois modelos

Certificação LPIC 1

Descrição	SysVinit	Systemd
RunLevel Halt	0	runlevel0.target, poweroff.target
Single User Mode	1	runlevel1.target, rescue.target
Multiusuário	2	runlevel2.target, multi- user.target
Multiusuário com Rede	3	runlevel3.target, multi- user.target
Run Level 4	4	runlevel4.target, multi- user.target
Multiusuário com interface gráfica e rede	5	Runlevel5.target, graphical.target
Shell emergência	emergency	emergency.target
Mudar run level	telinit 3	systemctl isolate runlevel3.target

Certificação LPIC 1

Descrição	SysVinit	Systemd
Configurar run level 3 para próximo boot	Editar /etc/inittab	In -sf /lib/system/system/multi-user.target /etc/systemd/system/default.target
Verificar o run level atual	runlevel	systemctl get-default

```
[root@dev ~]# cat intro.txt
```

- **As diferenças entre o SysVInit e Systemd não param por aí ...**
- **No modelo SysVInit o utilitário service controlava os serviços em execução ou não no sistema e com ele era possível para ou iniciar daemons etc. No caso de habilitar ou desabilitar serviço ficava a cargo chkconfig, mas no modelo systemd tudo isso ficou a cargo de uma única ferramenta**
- **Vamos apresentar mais alguns slides com os comparativos ...**

Certificação LPIC 1

Descrição	SysVinit	Systemd
Iniciar um serviço	<code>service serviço start</code>	<code>systemctl start serviço</code>
Para um serviço	<code>service serviço stop</code>	<code>systemctl stop serviço</code>
Reiniciar um serviço	<code>service serviço restart</code>	<code>systemctl restart serviço</code>
Recarregar um serviço	<code>service serviço reload</code>	<code>systemctl reload serviço</code>
Verificar status de um serviço	<code>service serviço status</code>	<code>systemctl status serviço</code>
Habilitar um serviço no boot	<code>chkconfig serviço on</code>	<code>systemctl enable serviço</code>
Desabilitar um serviço no boot	<code>chkconfig serviço off</code>	<code>systemctl disable serviço</code>
Verificar se um serviço está habilitado	<code>Chkconfig --list serviço</code>	<code>systemctl is-enable serviço</code>

Certificação LPIC 1

```
[root@dev ~]# cat intro.txt
```

- Até mesmo nas tarefas mais básicas como rebootar o sistema e verificar logs o systemd mudou a forma de realizar isso
- Observe que o systemd tenta concentrar a maior parte das funções de controle na ferramenta **systemctl ...**

Certificação LPIC 1

Descrição	SysVinit	Systemd
Desligar o sistema	halt	systemctl halt
Desligar um sistema com poweroff	poweroff	systemctl poweroff
Reiniciar um sistema	reboot	systemctl reboot
Suspender um sistema	pm-suspend	systemctl suspend
Hibernar um sistema	pm-hibernate	systemctl hibernate
Verificar logs do sistema	tail -f /var/log/messages ou tail -f /var/log/syslog	journalctl -f

[root@dev ~]# apropos SysVInit Systemd

- Os principais comandos para trabalhar com os dois modelos de controle de inicialização do sistema e de serviços:
 - service, chkconfig, ntsysv, dmesg, /var/log/messages
 - systemctl e journalctl

Certificação LPIC 1

[root@dev ~]# service postfix status

- **Instalação: [base]**
- **Descrição: Controla ou verifica o status de um serviço**
- **Ações possíveis:**
 - **stop**
 - **start**
 - **restart, reload**
 - **status**
- **Manpage: man service**


```
[root@dev ~]# chkconfig --list
```

- **Instalação: [base]**
- **Descrição: Habilita ou desabilita um serviço**
- **Ações possíveis:**
 - **--add: habilita um serviço**
 - **--del: desabilita um serviço**
 - **--level [0123456] serviço on off: indica em qual level o serviço deve ser habilitado ou desabilitado**
- **Manpage: man chkconfig**

```
[root@dev ~]# ntsysv
```

- **Instalação: yum install ntsysv**
- **Descrição: Habilita ou desabilita um serviço**
- **Ações possíveis:**
 - **[interface gráfica]**
- **Manpage: man ntsysv**

Certificação LPIC 1

[root@dev ~]# dmesg

- **Instalação: [base]**
- **Descrição: lista informações registradas pelo kernel durante inicialização do sistema**
- **Manpage: man dmesg**

Certificação LPIC 1

```
[root@dev ~]# tail -100 /var/log/messages
```

- **Instalação: [base]**
- **Descrição: Armazena informações registradas pelo kernel ou por serviços do sistema em tempo de execução**

```
[root@dev ~]# systemctl -t service
```

- **Instalação: [base]**
- **Descrição: Habilita, desabilita, para, inicia, reinicia etc.**
- **Ações possíveis:**
 - **start, stop: inicia ou para um serviço**
 - **enable, disable: habilita ou desabilita um serviço**
 - **-t service: lista todos os serviços**
- **Manpage: man systemctl**

[root@dev ~]# journalctl -f

- **Instalação: [base]**
- **Descrição: Verifica informações geradas/gerenciadas pelo systemd**
- **Principais parâmetros:**
 - **-f: igual ao tail -f (lê continuamente ou monitora o arquivo)**
 - **-k: imprime apenas as mensagens do kernel**
- **Manpage: man journalctl**

Certificação LPIC 1



INICIAR
FIM DE SEMANA