Revisão Completa dos Tópicos do Exame 201-450

Tópico 200 – Planejamento de Capacidade

200.1 – Medir e Resolver Problemas de Uso de Recursos

Comando iostat

Função: Relatório de uso de CPU e I/O de dispositivos de armazenamento

Uso e Principais Opções:

iostat : Mostra o relatório da CPU e de I/O, desde a inicialização do sistema
iostat -m : Informações em MB
iostat -h : Formato numérico simplificado (human)
iostat -c : Dados de CPU
iostat -d : Dados de I/O por Dispositivo (Device)
iostat -p : Dados de I/O por Partição
iostat <x> : Atualiza o relatório a cada x segundos
iostat <x> <y> : Atualiza o relatório a cada x segundos por y vezes

Tipos de consumidores de CPUs:

- %user – Aplicações a nível de usuário
- %nice – Aplicações a nível de usuário com definições de nice (alteração de prioridade)
- %system – Execuções no nível do sistema, kernel (tratamento de interrupts, gerenciamento de recursos de hardware, etc)
- %iowait - % do tempo em que as CPUs ficaram ociosas devido a leitura e escrita no disco
- %steal - % de tempo que uma CPU virtual aguardou pela CPU real enquanto o hypervisor do sistema estava servindo outra CPU virtual
- %idle - % de tempo que as CPUs ficaram realmente ociosas, desconsiderando as escritas de IO

Comando vmstat

Função: Alternativa ao iostat. Informações sobre processos, memória, paginação, I/O, Swap e CPU

Uso e Principais Opções:

vmstat : Mostra as informações desde a inicialização do sistema
vmstat -d : Estatísticas dos dispositivos de armazenamento (disco)
vmstat -p /dev/particao : Estatística de uma partição específica
vmstat <x> : Atualiza o relatório a cada x segundos
vmstat <x> <y> : Atualiza o relatório a cada x segundos por y vezes

Informações Importantes:

- A primeira coluna “r”, indica a quantidade de processos executando ou aguardando por execução (runnable processes)
- A segunda colulna “b”, indica processos em estado “ininterruptamente dormentes” (uninterruptible sleep / dormant). Processos podem ficar nesse estado enquanto aguardam um retorno de I/O em disco.

Comando mpstat

Função: Estatísticas detalhadas das CPUs

Uso e Principais Opções:

mpstat : Mostra as informações desde a inicialização do sistema
mpstat <x> : Atualiza o relatório a cada x segundos
mpstat <x> <y> : Atualiza o relatório a cada x segundos por y vezes

---

Comando sar

Função: Exibir informações estatísticas e históricas do uso de diversos recursos do sistema

Uso e Principais Opções:

sar : Uso das CPUs
sar -u : Uso das CPUs
sar -r : Uso da Memória
sar -S : Uso da área de Swap
sar -n DEV: Estatísticas de tráfego de todas as interfaces
sar -b : Estatísticas de I/O
sar -d : Uso de I/O por dispositivo
sar -B : informação sobre paginação
sar -f <arquivo> : Exibe as estatísticas de algum arquivo específico. Utilizado para ver dados de dias anteriores.

sar <x> : Atualiza o relatório a cada x segundos
sar <x> <y> : Atualiza o relatório a cada x segundos por y vezes

Informações Importantes:

- O processo/script sa1 é configurado na crontab e é responsável por coletar e armazenar os dados. O sa1 é uma variante do sadc (system activity data collector)
- Os registros são armazenados no diretório /var/log/sysstat (padrão Debian) ou /var/log/sa (padrão RedHat)

Comando netstat

Função: Exibir conexões de rede, sockets, tabelas de roteamento, estatísticas das interfaces e etc.

Uso e Principais Opções:

netstat -i : Lista as interfaces de rede e suas respectivas estatísticas de tráfego
netstat -s : Estatísticas de cada protocolo de rede
netstat -a: informações de todos os sockets, em estado listening ou não

---

Comando ss

Função: Traz informações semelhantes ao netstat, usado para verificar conexões de rede, socket, interfaces e etc

Uso e Principais Opções:

ss -s : Resumos das conexões e sockets
ss -l : Conexões em estado listen
ss -t -a: Exibe todos sockets TCP
ss -ltn : Exibe sockets TCP em estado listen. O -n faz com que os nomes não sejam “resolvidos”

Informações Importantes:

- O comando netstat busca as informações dos arquivos da partição virtual /proc enquanto que o ss obtém as informações diretamente do espaço de memória do kernel

---

Comando iptraf (ou iptraf-ng)

Função: Monitoração do tráfego pelas interfaces de rede em tempo real através de uma interface gráfica mas que não precisa de um servidor X, podendo ser utilizado via SSH.

Uso e Principais Opções:

- IP traffic monitor: Monitor do tráfego de pacotes em tempo real
- General interface statistics: Tráfego por interface de rede
- Detailed interface statistics: Tráfego detalhado por interface, mostrando detalhes por protocolo
- Statistical breakdown: Permite agrupar o volume de tráfego por tamanho dos pacotes e por porta

- LAN station monitor: Monitoração a nível de rede (ethernet) e MAC address
- Filters: Permite criar e gerenciar filtros para limitar a exibição dos pacotes

---

Comando ps

Função: Exibe informações detalhadas dos processos em execução

Uso e Principais Opções:

ps -ely | grep "^D" : Exibe todos os processos que estão em estado Dormant ou "Uninterruptible Sleep". É o mesmo estado exibido na segunda coluna do comando vmstat. Indica que o processo está aguardando pelo retorno de alguma atividade de I/O.

Informações Importantes:

- RSS (Resident Set Size) : Indica a quantidade de memória alocada na RAM para o processo
- VSZ (Virtual Memory Size) : Indica toda a memória que o processo tem acesso no sistema, inclusive o espaço de memória utilizado pelas bibliotecas compartilhadas que ele utiliza.

Comando pstree

Função: Exibe a árvore de processos e os relacionamentos entre processos pai e filho.

Uso e Principais Opções:

pstree : Exibe toda a árvore de processos
pstree -p : Exibe a árvore de processos incluindo os respectivos PIDs

---

Comando w

Função: Exibe os usuários logados e o que eles estão executando no sistema

---

Comando lsof

Função: Exibe os arquivos abertos permitindo associar esses arquivos a seus processos.

Uso e Principais Opções:

lsof -p PID : Exibe todos os arquivos abertos relacionados ao processo PID
lsof arquivo : Exibe o processo associado ao arquivo
lsof -i :80 : Exibe todas as conexões abertas na porta 80, através do arquivo aberto pelo socket.

Comando top

Função: Exibir os processos do sistema em uma interface que possibilita a monitoração em tempo real

Uso e Principais Opções:

- F (shift f) : Permite a administração dos campos, selecionados quais campos de cada processo serão exibidos. A opção “s” define o campo que ordenará a listagem.

Comando htop

Função: Possui a mesma função do comando top mas com recursos adicionais

Uso e Principais Opções:

htop -p PID : Exibe apenas as informações do processo especificado
htop -d 100 : Atualiza os dados a cada 10 segundos.

- Opção l : Lista os arquivos abertos pelo processo selecionado
- Opção h : Ajuda
- Opção H : Exibe/Oculta as threads dos processos

Comando iotop

Função: Monitor que exibe quanto cada processo está fazendo de I/O.

Comando uptime

Função: Exibe a quanto tempo o sistema está ativo, quantos usuários estão conectados e o load average.
Informações Importantes:

- O load average é uma métrica que indica, de maneira geral, a quantidade de processos em execução ou na fila de execução das CPUs. São exibidos 3 valores de médias, a cada 1, 5 e 15 minutos.

O Espaço de Swap

Informações Importantes:

- De maneira geral, a área de swap é um espaço em disco (através de um arquivo ou partição) que será utilizado caso o espaço de memória RAM seja completamente utilizado.
- É recomendado, mas não obrigatório, que um sistema Linux possua uma área de swap

- O uso constante da área de swap não é desejado visto que a leitura e escrita em disco é bem mais lenta que em memória RAM, fazendo assim com que os processos e o sistema tenha seu desempenho prejudicado.
- O uso constante da área de swap indica que alguma ação de capacidade deve ser feita no ambiente, como por exemplo, inclusão de novos servidores ao cluster, aumento de memória RAM, ajuste de configuração na aplicação e etc.
- O espaço alocado e utilizado da swap pode ser visto em comandos como cat /proc/swaps, free, entre outros.

200.2 – Prever Necessidades Futuras de Recursos

Softwares para análise de dados, monitoração e planejamento de crescimento:

collectd

Função: Daemon que coleta informações de recursos do sistema e os armazena em arquivos no formato RRD (Round Robin Database).

Site https://collectd.org/

Informações importantes: O collectd é baseado em Plugins e o arquivo de configuração basicamente determina quais serão utilizados e as principais opções sobre eles. O parâmetro que determina o carregamento de um plugin é o LoadPlugin.

RRDTool

Função: Ferramenta que coleta e armazena informações de recursos do sistema, gera relatórios gráficos e permite que estes dados sejam utilizados também por outras aplicações como o Cacti.

Site: https://oss.oetiker.ch/rrdtool/

Cacti

Função: Utiliza os dados gerados pelo RRDTool para gerar relatórios gráficos mais elaborados e em tempo real. Também permite a monitoração através do protocolo SNMP (Simple Network Management Protocol).

Site: www.cacti.net

MRTG (Multi Router Traffic Grapher)

Função: É focado na monitoração de tráfego de rede, apesar de também poder ser usado para recursos de hardware. Também pode monitorar via SNMP.

Site: https://oss.oetiker.ch/mrtg/

Nagios

Função: Solução completa de monitoramento centralizados de diversos elementos da rede, como computadores, servidores, roteadores, switchs, impressoras e etc. A coleta de dados é feita através de agentes ou protocolos como o SNMP. Um dos principais recursos é enviar alarmes aos administradores a partir dos dados coletados.

Site: https://www.nagios.com/

Icinga 2

Função: Assim como o Nagius, é uma solução completa de monitoramento e alarmes. De maneira geral possui as mesmas funções e modo de funcionamento.

Site: https://www.icinga.com/products/icinga-2/

Tópico 201 – Kernel Linux

201.1 – Componentes do Kernel

Conceitos

- Linux é o kernel, o núcleo do sistema operacional
- Tipos de Kernel:
  - Monolítico: Arquitetura em que toda a implementação do kernel é feita dentro de um processo único, através de uma imagem única.
  - Micro-Kernel: Nessa arquitetura o kernel é um controlador central que degela funções a outros subprocessos, ou servers.
  - O kernel Linux segue a arquitetura Monolítica, porém de forma modular. As implementações principais de gerenciamento são feitas na imagem do kernel, porém ele também trabalha em conjunto com módulos, que implementam recursos específicos e são carregados em memória apenas quando necessário.

As Versões do Kernel

- Versões disponibilizadas pelo site http://www.kernel.org
  - Indicações dos releases:
    - Mainline: árvore mantida pelo Linus Torvalds em que todas as novas funcionalidades são implementadas e desenvolvidas;
    - Stable: Quando o mainline é liberado (released), é considerado “stable”.
    - Longterm: Versões do kernel que são oficialmente mantidas. Correções importantes de bugs e segurança também são aplicadas nessas versões.
    - EOL (End of Life): Indica que em breve este release do kernel deixará de ser mantido.
- Atualmente a nomenclatura da versão do kernel segue o formato A.B.C, aonde:
  - A é a versão principal do kernel
  - B é a release principal, contendo novas implementações
  - C é a release secundária, ou minor release, indicando principalmente a correções
- Durante a versão 2.6 o padrão possuía 4 números, A.B.C.D, e o 2.6 (A.B) era considerado a versão estável

Os Arquivos e a Localização do Kernel e da Imagem

- Fontes do Kernel: /usr/src/linux/
- Documentação do kernel: /usr/src/linux/Documentation/
- Imagem do Kernel: /boot/ (arquivos vmlinuz*)
- Módulos: /lib/modules/<kernel>/ (arquivos *.ko)
- O arquivo tar com as **fontes** é compactado com o algoritmo **XZ**
- É indicado criar um link simbólico chamado “linux” apontando para a fonte da kernel atual em /usr/src
  - ln -s linux-A.B.C linux

- Há dois tipos de imagem de kernel: zImage ou bzImage
  - zImage possui o tamanho máximo de 512 Kb e é alocada na memória baixa (low memory)
  - bzImage (big zImage) não tem limite de tamanho e é alocada na memória alta (high memory)
  - Inicialmente ambas **imagens** eram compactadas com o **gzip,** porém desde o Linux 2.6.30, vários outros algoritmos são suportados, como, xz LZMA e o mais comum atualmente **bzip2**.

201.2 – Compilando um kernel

O processo é composto basicamente de 3 fases:

- Configuração
- Compilação
- Instalação

Configuração

O processo de configuração tem por objetivo gerar o **arquivo .config** que vai ser usado durante o processo de compilação. Esse arquivo indica os recursos que devem ser instalados, e quais devem ser construídos dentro do kernel ou implementados via módulo.

As opções disponíveis são:

- make config : são feitas perguntas sobre todos os recursos
- make oldconfig : O .config atual será lido e caso hajam opções não informadas, serão perguntadas.
- make menuconfig : Interface de menus através da lib ncurses
- make xconfig : Interface gráfica através da biblioteca QT
- make gconfig : Interface gráfica através da biblioteca GTK

Compilação

Nesta fase vamos compilar o kernel e os módulos associados, gerando o arquivo de imagem bzImage que será usado durante o boot pelo GRUB.

Os comandos utilizados são:

- make bzImage : Compilar o código fonte e gerar o arquivo de imagem bzImage em /usr/src/linux/arch/x86/boot/bzImage
- make modules : Compilar os módulos

Instalação

Para instalar os módulos do novo kernel utilizamos o comando “make modules_install”. O comando criará um novo diretório /lib/modules/<versao-release-kernel> com os módulos da versão atual e os arquivos modules.*, criados pela execução do depmod.

A instalação da nova imagem do kernel pode ser feita de 2 formas:

- Pelo comando “make install” que automaticamente copiará o bzImage para o /boot, e fará o mesmo com os arquivos .config e System.map. Além disso fará a geração do arquivo initrd. E por final executará o update-grub para atualizar as configurações do GRUB.

- Manualmente: Simplesmente copiando o arquivo bzImage para o diretório /boot e renomeando-o. Utilizando o mkinitramfs/mkinitd/update-initramfs para gerar o novo arquivo Initial Ramdisk. E, ao final, executando o update-grub para atualizar o GRUB.

Limpeza

Durante o processo de compilação e instalação de um novo kernel, podem ocorrer erros que exijam que todo o processo seja reiniciado. Nesses casos é importante o uso de comandos que removam os arquivos criados durante o processo. Além disso, depois que o kernel foi compilado e instalado, os objetos criados na árvore /usr/src não são mais necessários e podem ser limpos para liberar espaço. Os principais comandos são:

- **make clean :** Remover os objetos criados durante o processo de compilação, por exemplo arquivos com a extensão .o e .ko.
- **make mrpropper :** Além do que já é limpo pelo “make clean” remove também o arquivo de configuração .config.
- **make distclean:** Nesta opção, além de ser removido tudo que já é removido pelo mrpropper, também são apagados arquivos de backup, temporários, dumps, ou seja, tudo que não é disponibilizado originalmente com o código fonte.

Geração do Initial Ramdisk

O Initramfs (ou Initrd) é um arquivo de imagem que será utilizado durante o processo de boot. Essa imagem será montada na memória RAM e utilizada como uma partição raiz (/) temporária, que conterá por exemplo os módulos que precisam ser usados pelo kernel, por exemplo um controlador RAID, de uma interface de rede e etc.

Vale mencionar que o initramfs é um arquivo originalmente compactado com o **cpio** e em algumas distribuições também compactado com o **gzip**.

Em sistemas padrão Debian, são utilizados os comandos:

- mkinitramfs
  - # mkinitramfs -o /boot/iniramfs-x.y.z x.y.z
- update-initramfs
  - update-initrafms -u (atualizar um arquivo já existente)
  - update-initramfs -c (criar um novo arquivo)

Em sistemas padrão Red Hat, são utilizados os comandos

- mkinitrd (atualmente chama o dracut)
  - # mkinitrd -c -o /boot/initrd.img-x.y.z -k x.y.z
- dracut
  - # dracut
  - # dracut --force (caso já exista um arquivo)
  - # dracut novo-arquivo.img (gerar um novo arquivo)

Empacotamento do kernel

make rpm-pkg : gera um conjunto de pacotes .rpm
make binrpm-pkg : gera apenas o pacote rpm com a imagem compilada
make deb-pkg : gera um conjunto de pacotes .deb
make tar-pkg : gera um arquivo tar sem compressão
make targz-pkg : gera um arquivo tar com compressão gzip
make tarbz2-pkg gera um arquivo tar com compressão bzip2

DKMS (Dynamic Kernel Module Support)

O DKMS é um framework que permite integrar módulos cujas fontes ficam fora da árvore de fontes do kernel.

A principal função é que esses módulos sejam automaticamente reconstruídos sempre que uma nova versão do kernel for instalada.

Para fazer essa integração, nas fontes do módulo deve ser configurado o arquivo dkms.conf.

201.3 – Gerenciando e Resolvendo Problemas no Kernel em Tempo de Execução

Comando uname

Função: Exibe informações referentes ao kernel em uso.

Uso e Principais Opções:

uname -r : Mostra a release do kernel em uso
uname -a : Mostra todas as informações do kernel em uso

Visualização e Alteração de Parâmetros do Kernel

Pode ser feito diretamente nos arquivos do /proc. Exemplo:

cat /proc/sys/fs/file-max
echo 100000 > /proc/sys/fs/file-max

Ou pelo uso do comando sysctl:

sysctl fs.file-max
sysctl -w fs.file-max=100000

Para que a configuração seja permanente, ele deve ser feita nos seguintes arquivos ou diretórios:

- /etc/sysctl.conf
- /etc/sysctl.d/

Os comandos a seguir também são utilizados para obter informações do sistema, através das referências do /proc:

- lspci – Dispositivos conectados ao barramento PCI
  - # lspci -s 00:11 -v
- lsusb – Dispositivos conectados ao barramento USB
  - # lsusb -d xxx:yyy -v
  - # lsusb -s 00:11 -v
- lsdev – Informações dos hardwares utilizados

O **dmesg** é o comando utilizado para acessar e controlar as informações do "**kernel ring buffer"**, que entre outros dados, contém as informações do boot do sistema.

Módulos

Os módulos utilizados pelo kernel ficam em /lib/modules/`uname -r`.

Principais comandos relacionados ao gerenciamento de módulos:

- lsmod : lista os módulos carregados
- modinfo : Exibe informações detalhadas de um módulo
  - # modinfo psmouse
  - # modinfo -p psmouse (exibe apenas os parâmetros do módulo)
- insmod : carrega um módulo mediante a indicação do nome do arquivo .ko do módulo
- rmmod : descarrega um módulo
- modprobe : pode ser utilizado para carregar e descarregar um módulo a partir do seu nome. O modprobe utiliza as informações dos arquivos /lib/modules/`uname -r`/modules.* para relacionar os nomes aos arquivos, identificar dependências e etc.
  - # modprobe psmouse (carrega o módulo)
  - # modprobe -r psmouse (descarrega o módulo)
  - # modprobe modulo parametro=valor (define um parâmetro de um módulo)

Arquivos de Configuração:

- /etc/modules.conf : Configurações dos módulos
- /etc/modprobe.d/* : Configurações dos módulos
- /etc/modules : Módulos que devem ser carregados no boot do sistema
- /etc/module-load.d/* : Módulos que devem ser carregados no boot do sistema

Principais opções que podem ser utilizadas nos arquivos de configuração:

- options : Especificar uma opção para um módulo
  - # options r8169 debug=16
- alias : Definir um outro nome para um módulo
  - # alias nome-alias nome-modulo
- install : Definir ações que serão executadas para o carregamento do módulo.
  - # install moduloX /sbin/modprobe --ignore-install moduloX && /bin/touch /tmp/moduloX.tmp
- remove : Definir ações que serão executadas para o descarregamento de um módulo
  - # remove moduloX /sbin/rmmod moduloX && /bin/rm /tmp/moduloX.tmp
- * Quando o modprobe é usado nas opções install e remove, devem ser usados os parâmetros --ignore-install e --ignore-remove, para que o modprobe não entre em um loop.

Udev (Dynamic Device Management)

Processo responsável por gerenciar em tempo real as referências em /dev/ e /sys/. O processo roda em background ouvindo as notificações do kernel sobre novos dispositivos de hardware ou remoção deles.

Principais arquivos de configuração:

- /etc/udev/udev.conf : Arquivo de Configuração
- /etc/udev/rules.d/* : Arquivos com as Regras

Principais comandos utilizados:

- udevmonitor (ou)
- udevadm monitor

Tópico 202 – Inicialização do Sistema

202.1 – Customizando o Sistema de Inicialização SysV-init (e systemd)

SysV – Init

Funcionamento: O SysV utiliza o conceito de runlevels que definem os grupos de aplicações que serão iniciados em cada nível, sendo eles:

- 0 = Desligamento (Halt)
- 1 (s) = Modo Mono-Usuário (single user), ou de manutenção
- 2 - 5 = Modos Multi-Usuários, com variações conforme a distribuição
- 6 = Reboot

Principal Arquivo de Configuração:

- /etc/inittab

Configuração do Runlevel Default no arquivo /etc/inittab:

- id:2:initdefault:

Localização dos Scripts de Inicialização:

- /etc/init.d/

Diretórios dos Runlevels:

- /etc/rc*.d/
- Cada diretório contém links para os scripts presentes no diretório /etc/init.d/, indicando quais processos devem ser iniciados ou mortos.

Principais Comandos e Opções:

- runlevel : Exibe o runlevel utilizado anteriormente e o atual
- init / telinit : Troca para outro runlevel
- update-rc.d : Gerencia os runlevels e os links do diretório /etc/rc*.d/ em sistemas padrão Debian
  - # update-rc.d process disable 4 : Desabilita o serviço “process” do runlevel 4
  - # update-rc.d process enable 4 : Habilita o serviço “process” no runlevel 4
  - # update-rc.d process remove : Remove todas as configurações do serviço “process”
    - -f: A opção -f força a remoção de todos os links simbólicos do /*etc/*rcX.d/, mesmo que o script do serviço ainda ainda exista no /etc/init.d/
  - # udpate-rc.d process defaults : Restaura as configurações default para o serviço “process”
- chkconfig : Gerencia os runlevels e os links do diretório /etc/rc*.d/ em sistemas padrão RedHat
  - # chkconfig : mostra as configurações atuais
  - # chkconfig --list process : Exibe as configurações do serviço “process”
  - # chkconfig --level 45 “process” off : Desabilita o serviço “process” dos runlevels 4 e 5
  - # chkconfig --level 4 “process” on : Habilita o serviço “process” no runlevel 4

systemd

Funcionamento: Ao invés de trabalhar com runlevels, o systemd trabalha com Units e Targets. Uma unidade é identificada como nome.tipo, possuindo então um arquivo de configuração. Target é um tipo de Unit que agrupa várias Units, sendo de certa forma correspondente ao conceito de Runlevel usado no SysV.

É importante mencionar que o systemd mantém compatibilidade com o SysV e os scripts init no padrão LSB, que por exemplo definem os “runlevels” default de cada serviço.

Diretórios em que se localizam os arquivos de configuração das Units:

- /etc/systemd/system
- /run/systemd/system
- /lib/systemd/system

Os arquivos de configuração podem estar localizados em qualquer dos diretórios acima, mas seguem a seguinte ordem de prioridade: /etc/systemd tem maior prioridade, seguido do /run/systemd e o /lib/systemd tem a menor prioridade.

O target Default, que será iniciado após o processo de boot, é definido pelo arquivo default.target, que em geral é um link simbólico para algum dos targets do sistema.

Principais Comandos e Opções:

- systemd-delta : Analisa os 3 diretórios de configurações e indica quando há arquivos iguais se sobrepondo ou até se completando.
- systemctl : Principal comando de gerenciamento dos serviços do systemd.
  - list-units : Lista todas as Units do sistema
  - default : Entra no target default
  - isolate “target” : Entra em um target específico
  - stop / start / status “process” : Realiza ações em um serviço específico
  - enable / disable “process” : Define se um processo deve ou não ser iniciado após o boot
  - daemon-reload : Re-lê as configurações do daemon do systemd

---

LSB (Linux Standard Base)

Conjunto de especificações que visam definir um padrão para as diversas distribuições Linux, entre elas:

- Bibliotecas instaladas por padrão
- Comandos
- Hierarquia do sistema de arquivos
- Níveis de Execução
- etc

202.2 – Recuperação do Sistema

GRUB 2 e GRUB Legacy

- O GRUB 2 é uma nova implementação do Bootloader GRUB e é atualmente o padrão utilizado
- O GRUB 2 é modular, suporta scripts, pode ser utilizado tanto com firmware BIOS quanto UEFI, suporta múltiplas plataformas, além de diversas outras características que o tornaram muito mais flexível e moderno.

Tecnicamente, a tabela abaixo contém as principais diferenças entre as duas implementações:

	GRUB LEGACY	GRUB 2
Arquivos de Configuração	/boot/grub/menu.lst	/boot/grub/grub.cfg /etc/default/grub /etc/grub.d/
Referência ao disco	hda1 = hd0,0 hda5 = hd0,4 hdb3 = hd1,2	hda1 = hd0,1 ou hd0,msdos1 hda5 = hd0,5 ou hd0,gpt5 hdb3 = hd1,3
Comandos	# grub-install /dev/sda # grub-install '(hd0)'	# grub-install <device> # update-grub # grub-mkconfig -o /boot/grub/grub.cfg
Principais parâmetros	title "Ubuntu" **root (hd0,0)** **kernel** /boot/vmlinuz-4-8.0-46-generic ro root=/dev/sda5 mem=4096M initrd /boot/initrd-4-8.0-46-generic default=0 timeout=15	menuentry "Ubuntu" **{** **set root=(hd0,1)** **linux** /boot/vmlinuz-4-8.0-46-generic ro root=/dev/sda5 mem=4096M initrd /boot/initrd-4-8.0-46-generic **}** GRUB_DEFAULT=0 GRUB_TIMEOUT=15

BIOS e UEFI

- BIOS = Basic Input Output System
- UEFI = Unified Extensible Firmware Interface
- BIOS e UEFI são duas implementações diferentes para o firwmare de um sistema
- A BIOS carrega o bootloader através da área de MBR (Master Boot Record)
- O UEFI obtém os bootloaders específicos através do ESP (EFI System Partition), uma partição do tipo FAT normalmente montada no diretório /boot/efi/
- O UEFI implementa o recurso de Boot Seguro, em que antes de realizar o boot ele verifica se as imagens estão devidamente assinadas
- Dentro da partição ESP, os bootloaders podem ser reconhecidos pela "extensão" .efi
- No Linux, o UEFI pode ser verificado e configurado pelo comando "*efibootmgr*"

NVMe (Non-Volatile Memory Express)

- Usado para discos SSD (Solid-State Drive)
- Interface criada pela Intel para que o os discos SSDs sejam usados no barramento PCI Express
- Em sistemas Linux é mapeado como /dev/nvme*

Procedimentos de Recuperação do Sistema

Boot em Modo Single (Mono-Usuário)

- Na tela inicial do GRUB, usar a tecla “e” para editar uma das entradas
- Incluir no parâmetro “linux”, uma das opções: 1, s, single
- Pressionar Ctrl X ou F10 pra realizar o Boot
- Informar a senha de root para entrar no modo de manutenção
- Nesse modo, todas as partições serão montadas normalmente e o administrador pode realizar as devidas configurações.
- Também poderá desmontar as partições visto que não haverá serviços em execução. Com as partições desmontadas pode realizar por exemplo checagens do disco com o uso do comando fsck.
- Após a manutenção de uma partição, ela pode ser remontada, como leitura e escrita, através do comando abaixo:
  - mount -o remount,rw /

Boot Direto no Bash

- Na tela inicial do GRUB, usar a tecla “e” para editar uma das entradas
- Incluir no parâmetro “linux”, a opção “init=/bin/bash”
- Pressionar Ctrl X ou F10 pra realizar o Boot
- Após o boot será iniciado diretamente o bash, ao invés do init (SysV/systemd)
- Nesse modo, as partições não são montadas, e os procedimentos de manutenção de disco e partições podem ser realizados

Uso do GRUB Shell

- Na tela inicial do GRUB, usar a tecla “c”
- O GRUB Shell permite que o administrador verifique as informações das partições do sistema e dê instruções para que seja realizado um novo boot
- Alguns dos comandos que podem ser utilizados:
  - **ls**
  - **ls -l** (hd0,msdos1)/
  - **set root=**(hd0,msdos1) : definir a partição em que se encontra o diretório /boot
  - **linux** /vmlinuz-X.Y.Z root=/dev/sda1 : definir a imagem a ser usada durante o boot e em qual partição o diretório / deve ser montado
  - **initrd** /initrd.img-X.Y.Z : definir a imagem de InitRD que deve ser utilizada
  - **boot** : realizar o boot pelas configurações definidas manualmente no GRUB Shell

Boot através de CD/DVD Live

- Iniciar o sistema através de um CD/DVD Live ao invés dos discos do sistema
- Útil quando os bootloaders como GRUB e a área da MBR possuem erros ou algum tipo de corrompimento.
- O comando “grub-install /dev/sdX” pode ser usado para reinstalar o GRUB
- O comando “update-grub” pode ser utilizado para corrigir e atualizar as configurações do GRUB

202.3 – Carregadores de Boot Alternativos

SYSLINUX

- Bootloader Linux criado para trabalhar com filesystems do tipo FAT (MS-DOS)
- Usado por exemplo em pendrives de recuperação
- Com o tempo foram adicionados ao projeto o suporte a outros bootloaders:
  - EXTLINUX
  - ISOLINUX
  - PXELINUX

EXTLINUX

- Usado em sistemas de arquivos nativos do Linux (ext*, brtfs e xfs)
- Obtém as informações do diretório /boot/extlinux
- Configuração em extlinux.cnf

ISOLINUX

- Carregador de boot para CD-ROMs com filesystem ISO 9660
- Usado em LiveCDs e LiveDVDs
- Armazenado normalmente no diretório /boot/isolinux/
- O arquivo de bootloader é o isolinux.bin e o de configuração isolinux.cfg

---

PXELINUX

- PXE : Pre-Boot Execution Environment
- Usado para Boot via interface de rede
- O carregador de boot (pxelinux.0) é enviado via rede ao cliente
- A placa de rede deve suportar o recurso
- No cliente a configuração é feita na BIOS ou UEFI
- O PXE Server também deve conter um DHCP e um TFTP
- Arquivos disponibilizados pelo TFTP
  - /tftpboot/pxelinux.0
  - /tftpboot/pxelinux.cfg

Systemd-boot

- Bootloader do “pacote” systemd
- Criado para funcionar com firmware UEFI

U-Boot

- The Univesal Boot Loader
- Usado em sistemas embarcados (embedded)
- Suporta múltiplas arquiteturas

UEFI Secure Boot

- O UEFI suporta o Boot Seguro
- O Boot Seguro faz com que o UEFI só carregue imagens digitalmente assinadas
- O Linux carrega o bootloader **shim.efi** para lidar com os certificados e chaves, e então encaminha o fluxo para o **grubx64.efi** (padrão)

Tópico 203 – Sistemas de Arquivos e Dispositivos

203.1 – Operando o Sistema de Arquivos Linux

Configurando o /etc/fstab

Cada entrada no arquivo /etc/fstab é composta por 6 campos:

1. Dispositivo (device): Pode ser identificado pelo nome do device, como /dev/sdb1, pelo UUID ou pelo LABEL.
2. Ponto de Montagem (mountpoint): Diretório que será associado ao dispositivo. No caso de swap pode ser identificado como “none” ou “swap”
3. Tipo de FileSystem: Por exemplo ext4, vfat, ou auto para identificação automática
4. Opções: Opções de montagem
5. Dump: Determina se será feito um dump na partição. Em desuso.
6. Checagem do Disco: Determina se a partição será verificada pelo fsck no boot do sistema. 0 para não, 1 para a partição raiz, 2 para as demais partições.

O comando mount

Principais formas de uso:

- mount : lista as partições montadas no momento. A mesma informação pode ser obtida pelos arquivos:
  - /etc/mtab
  - /proc/mounts
- mount -t <tipo-filesystem> : lista apenas as partições do filesystem especificado
- mount <device> <mountpoint> : monta um device em um diretório
- mount -t <tipo-filesystem> <device> <mountpoint> : monta um device em um diretório usando um tipo específico
- mount <device> : faz a montagem de acordo com as configurações do /etc/fstab
- mount <mountpoint> : faz a montagem de acordo com as configurações do /etc/fstab
- mount -a : monta todos os dispositivos configurados com a opção “auto” no /etc/fstab
- mount -U <UUID> / mount UUID=<UUID> : monta um dispositivo pelo seu UUID
- mount -o opções : faz a montagem usando um grupo específico de opções

As **principais opções do mount,** utilizadas também no /etc/fstab:

- rw : Habilitar leitura e escrita na partição
- ro : Habilitar apenas leitura na partição
- auto : Montagem durante o boot e pela opção -a do comando mount
- noauto : Não monta automaticamente
- user : Permite que qualquer usuário possa montar a partição, mas apenas quem montou pode desmontar
- users: Permite que qualquer usuário possa montar e desmontar a partição.
- async / sync : Define se a gravação em disco será feita de forma síncrona ou assíncrona
- owner : Permite que o usuário dono do dispositivo possa montá-lo
- remount : Usado para remontar uma partição já montada, mas com outras opções

- uid / gid : Define o usuário ou grupo que será dono do diretório montado. Disponível apenas em alguns tipos de filesystems como fat e ntfs
- defaults: Habilita as seguintes opções: rw, suid, dev, exec, auto, nouser e async

---

Outros Comandos Úteis

- **blkid** : Mostra informações de todas as partições do sistema, incluindo UUID, Label e Tipo do Filesystem
- lsblk -f : Mostra informações dos discos e partições, incluindo tamanho, UUID, Label, Tipo do Filesystem e Ponto de Montagem
- e2label <device> : Mostra o Label de uma partição
- findfs UUID=<UUID> / LABEL=<LABEL> : Mostra o dispositivo através de seu UUID ou Label
- findmnt : Exibe uma relação de todas as partições montadas.
- **sync** : Efetiva a gravação de dados que estão em cache/buffer da memória para o disco
- umount: Usado para desmontar as partições. Segue o mesmo padrão de uso do comando mount.

Criação e Montagem da Área de Swap

Usando Partições:

- # mkswap /dev/sdX
- # swapon /dev/sdX
- Verificar: # swapon -s ou # cat /proc/swaps
- Configuração no /etc/fstab
  - /dev/sdX swap swap defaults 0 0
  - /dev/sdX none swap sw 0 0
- # swapon -a : Inclui na área de swap as partições/arquivos definidos no /etc/fstab

Usando Arquivos:

- # dd if=/dev/zero of=/arquivoswap bs=1024K count=1000
- # mkswap /arquivoswap
- # swapon /arquivoswap
- Configuração no /etc/fstab
  - /arquivoswap swap swap defaults 0 0
  - /arquivoswap none swap sw 0 0

Systemd Mount

O ponto de montagem pode ser configurado através de uma Unit no Systemd.

Para isso basta criar um arquivo do tipo ponto-de-montagem.mount nos diretórios de configuração do systemd (/lib/systemd/system ou /etc/systemd/system/).

A seguir um exemplo para o arquivo opt-teste.mount:

```
[Unit]
Description=Ponto de Montagem Teste

[Mount]
What=/dev/sdb1
Where=/opt/teste
Type=ext4
Options=defaults

[Install]
WantedBy=multi-user.target
```

Após a configuração é importante fazer um reload no systemd pelo comando:

- # systemctl daemon-reload

Dessa forma a partição pode ser administrada pelo comando systemctl:

- # systemctl stop opt-teste.mount
- # systemctl start opt-teste.mount
- # systemctl enable opt-teste.mount

203.2 – Manutenção de um Sistema de Arquivos Linux

Comandos mkfs, mkfs.* e mke2fs

Função: Comandos utilizados para criar a estrutura de um filesystem específico em um dispositivo.

Uso e Principais Opções:

mkfs -t <tipo-fs> <device> : Exemplo: # mkfs -t ext4 /dev/sdb1
mkfs.<tipo-fs> <device> : Exemplo: # mkfs.ext4 /dev/sdb1
mke2fs -t <ext*> <device> : Cria uma partição ext*, se o -t não for utilizado, cria como ext2.
mkfs -c … : A opção opção "-c" faz com que seja realizada uma checagem por badblocks no device antes da criação do fs.

Informações Importantes:

- O comando mkfs é na verdade um frontend que chama os mkfs.* específicos de cada tipo de filesystem
- O comando mke2fs é voltado para filesystems ext2/ext3/ext4
- Em resumo, um sistema de arquivos é um conjunto de estruturas lógicas e funções que determinam como os dados serão armazenados em um dispositivo

---

Comando dumpe2fs

Função: Utilizado para verificar os parâmetros e propriedades de uma partição que utiliza ext2/ext3/ext4.

Uso e Principais Opções:

dumpe2fs <device> : Exibe todas as informações do dispositivo, inclusive os grupos de blocos
dumpe2fs -h <device> : Exibe apenas as parâmetros de um dispositivo
dumpe2fs -b <device> : Exibe os badblocks, caso existam

Comando tune2fs

Função: Utilizado para modificar e ajustar (tunar) os parâmetros de um filesystem ext2/ext3/ext4

Uso e Principais Opções:

tune2fs -l <device> : Lista os parâmetros
tune2fs -L "Label" <device> : Device um nome ou label para o dispositivo
tune2fs -c999 <device> : Define a quantidade máxima de montagens para que o filesystem seja novamente verificado pelo fsck
tune2fs -i30d(w/m) <device> : Define o período máximo de tempo para que o filesystem seja novamente verificado. Pode ser definido em dias (d, padrão), semanas (w) ou meses (m)
tune2fs -mX <device> : Define a %X de blocos reservados para uso apenas de processos privilegiados.

tune2fs -j <device> : Inclui o atributo de journalling a um filesystem do tipo ext2, transformando-o em ext3

Comando debugfs

Função: Realizar alteração de baixo nível em um sistema de arquivos. Um exemplo de uso é recuperar arquivos apagados em sistemas ext2 e ext3.

Uso e Principais Opções:

debugs <device> : Entra em modo de prompt de comando para realização dos procedimentos.

Comando badblocks

Função: Procurar por badblocks em uma partição.

Uso e Principais Opções:

badblocks <device>
badblocks <device> -o arquivo.txt : Resultado será escrito no arquivo.txt

---

Comandos fsck, fsck.* e e2fsck

Função: Comandos utilizados para checar e reparar um filesystem.

Uso e Principais Opções:

fsck -t <tipo-fs> <device> : Exemplo: # fsck -t ext4 /dev/sdb1
fsck.<tipo-fs> <device> : Exemplo: # fsck.ext4 /dev/sdb1
fsck <device> : Dessa forma o tipo do filesystems será identificado automaticamente
e2fsck -t <ext*> <device> : Verifica uma partição do tipo ext*, se o -t não for utilizado, irá identificar automaticamente
fsck -y … : Responde sim para todas as confirmações de correção
fsck -n … : Responde não para todas as confirmações de correção. Na prática apenas verifica um filesystem mas não o corrige.
fsck -f … : Força a execução mesmo que o filesystem esteja indicado como “clean”
fsck -p … : Realiza todos os procedimentos sem pedir nenhuma confirmação do usuário

Informações Importantes:

- Assim como o mkfs, o fsck é um frontend para os fscks específicos de cada filesystem
- O comando e2fsck é voltado para fs do tipo ext2/ext3/ext4

Retornos do fsck:

- 0: Nenhum erro
- 1: Erros encontrados e corrigidos
- 2: O sistema deve ser reiniciado

- 4: Erros encontrados e deixados sem correção
- 8: Erro operacional
- 16 : Erro de uso ou sintaxe
- 128 : Erro de biblioteca compartilhada

BTRFS – Btree File System

- Foco em Tolerância a Falhas e Facilidade de Administração e Reparo
- Implementa RAID 0, 1 e 10 (5 e 6 em desenvolvimento)
- Possibilita a criação de SubVolumes e Snapshots
- Utiliza gravação por CoW (Copy-on-Write)

Comandos - Uso e Principais Opções:

btrfs filesystem show : Exibe informações de partições utilizando btrfs
btrfs subvolume create /ponto/montagem/Subvolume1 : Cria um Subvolume em /ponto/montagem
btrfs subvolume list /ponto/montangem [-t] : Lista os subvolumes criados em /ponto/montagem
btrfs subvolume show /ponto/montagem/Subvolume1 : Exibe detalhes do Subvolume1

mount -o subvol=Subvolume1 /dev/sdb6 /outra/montagem : Monta o Subvolume1 do /dev/sdb6 em outro diretório

btrfs subvolume snapshot /ponto/montagem/Subvolume1 /ponto/montagem/Subvolume1/Snap : Cria um snapshot do Subvolume1

mkfs.btrfs -d raid0/1/10 /dev/sdb1 /dev/sdb2 -f : Faz com que os devices /dev/sdb1 e /dev/sdb2 componham um dispositivo de RAID.

btrfs-convert <device> : Converte um dispositivo de ext* para btrfs

---

XFS

- Desenvolvido pela Silicon Graphics para o IRIX
- Robusto e Escalável
- Trabalha bem com arquivos grandes
- Redimensionamento/Desfragmentação Online
- Tamanho de Blocos Variáveis
- Utiliza Journaling

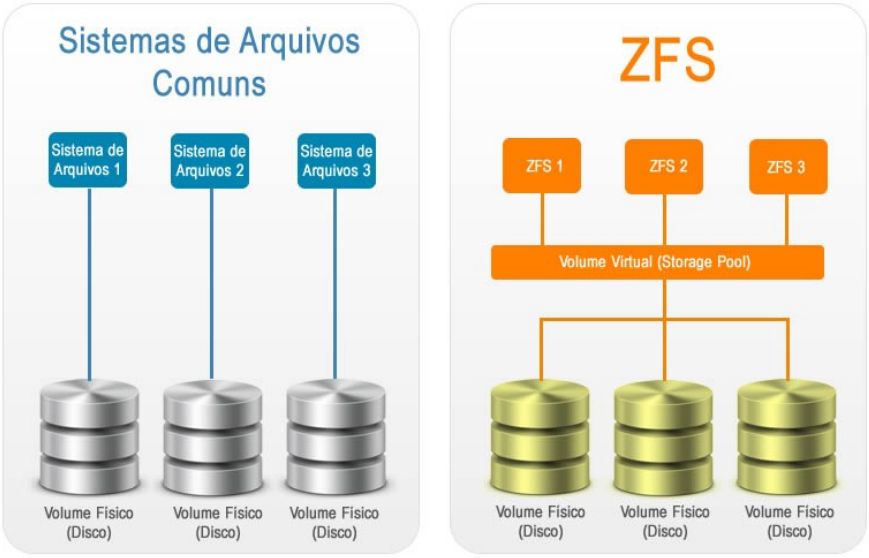
Comandos - Uso e Principais Opções:

xfs_info : Exibe informações do filesystem XFS
xfs_check : Checar por erros e/ou inconsistências no filesystem. Substituído pelo xfs_repair.
xfs_repair : Checa e Repara erros e/ou inconsistências em filesystems XFS

xfsdump : Cria um dump da partição para ser utilizada para backup
xfsrestore : Restaura um dump criado pelo xfsdump

ZFS (Noções)

- OpenZFS – Alternativa de Código Aberto
- Foco na Integridade dos Dados
- Facilidade de Gerenciamento
- Sistemas de Arquivos em Pools
- Escalabilidade
- Implementa recursos RAID
- Utiliza CoW (Copy-on-Write)
- Principais comandos:
  - zfs
  - zfspool



SMART (Self-Monitoring, Analysis ad Reporting Technology)

Função: Utilizado para verificar a saúde dos discos, erros nos dispositivos e controladoras que não são corrigidos pelo fsck. Utilizado apenas para identificar os erros, que na grande maioria dos casos não pode ser corrigido.

- Daemon: **smartd**
- Configuração: /etc/smartd.conf
- Utilitário: **smartctl**
  - # smartctl -a : Exibe todas as informações para o dispositivo, inclusive testes
  - # smartctl -i : Exibe apenas as informações do dispositivo
  - # smartctl -H : Exibe o resultado do teste de “saúde” (Health) do dispositivo

203.3 – Criando e Configurando Opções dos Sistemas de Arquivos

Montagem Automática via autofs

O recurso de montagem automática permite que uma partição só seja montada quando ela for realmente utilizada, assim como ela será desmontada após um tempo de inatividade.

Uma das formas de implementar esse recurso é através do **autofs**, pelo comando automount.

O processo **automount** roda como um daemon, monitorando os diretórios especificados e realizando a montagem quando forem utilizados.

Arquivos de Configuração para o autofs:

- /etc/auto.master (Master Map) : Arquivo em que são informados os diretórios que devem ser monitorados pelo automount. Exemplos:
  - /mnt/autofs /etc/auto.exemplo --timeout 30
  - /home /etc/auto.home
- /etc/auto.xxxxx : Arquivo de configuração específica de cada ponto de montagem automática. Exemplos:
  - exemplo -fstype=auto :/dev/sdb5
  - usuario -fstype=nfs 10.0.0.50:/dev/sdb5

Montagem Automática via systemd

O systemd também possibilita a criação de Units para a montagem automática. Para isso devem ser criados duas unidades, um .mount e um .automont, conforme o exemplo a seguir:

```
# cat mnt-systemdauto.mount
[Unit]
Description=Exemplo

[Mount]
Where=/mnt/systemdauto
What=/dev/sdb5

[Install]
WantedBy=multi-user.target

# cat mnt-systemdauto.automount
[Unit]
Description=Exemplo

[Automount]
Where=/mnt/systemdauto

[Install]
WantedBy=multi-user.target
```

Para que o systemd leia as novas unidades deve ser utilizado o comando “systemctl daemon-reload”

Criação de Imagens e Gravação de Cds/DVDs

Para criar uma imagem a partir de um CD/DVD:
dd if=/dev/sr0 of=imagem.iso

Para gravar uma imagem em um CD/DVD:
cdrecord dev=/dev/sr0 imagem.iso

Gerando Imagens a Partir de Arquivos e Diretórios

Feito através do comando **mkisofs** (genisoimage). Por padrão, a imagem é gerada usando o filesystem ISO 9660.

Exemplos e Opções de Uso:

Gerar uma imagem contendo todos os arquivos dos diretórios informados:
mkisofs -o imagem.iso /home/usuario /etc/config/

Gerar uma imagem contendo de maneira separada os arquivos de cada um dos diretórios informados:
mkisofs -o imagem.iso -graft-point dir1=/home/usuario dir2=/etc/config

O uso do ISO 9660 traz uma série de limitações na geração das imagens, principalmente relacionada à compatibilidade entre sistemas operacionais. Entre esses limitações temos:

- Uso do padrão de nomes 8.3, por exemplo, arquivos.pdf
- Limitações relacionados aos caracteres especiais (Unicode)
- Não mantém permissões e propriedades
- Não copia links simbólicos

Para solucionar essas limitações foram desenvolvidas extensões aos ISO 9660, entre elas:

- Joliet, opção -J
  - Habilita nomes longos, não ficando restritos ao formato 8.3
  - Suporte à codificação Unicode, possibilitando espaços e caracteres especiais
  - Indicado para quando o CD deva ser lido tanto em sistemas Linux como Windows
- Rock Ridge, opção -R
  - Além dos recursos do Joliet, possibilita a cópia de links simbólicos
  - Preserva permissões e propriedades dos arquivos

Além disso, o mkisofs tem as seguintes opções:

- El Torito
  - Utilizado para criar CDs Bootáveis
- HFS (-hfs)
  - Habilitar o uso do CD em sistemas Macintosh, que utiliza filesystems do tipo HFS
- UDF (-udf)
  - Cria a imagem usando o filesystem UDF, ao invés do ISO 9660

- O UDF (Universal Disk Format) visa ser um padrão único, que engloba os padrões de nomes e atributos de todos os sistemas operacionais
- Muito utilizado para criação de DVDs

Criptografia do Sistema de Arquivos

Os dados de um sistema de arquivos podem ser totalmente criptografados de forma que apenas através de uma senha ele possa ser montado para uso.

Para esse processos, são utilizados os seguintes recursos e ferramentas:

- LUKS (Linux Verified Key Setup) : Método de encriptação de dados
- dm-crypt : Sistema que faz parte do Device Mapper para realizar a criptografia dos devices
- Device Mapper : framework que cria devices virtuais para mapear blocos físicos, muito utilizado pro exemplo com LVM e RAID
- cryptsetup : Comando que irá utilizar os recursos do dm-crypt e LUKS para criar e gerenciar os devices criptografados

Processo de Criação e Montagem de um Dispositivo Criptografado:

- Preparar o device e definir uma senha:
  - # cryptsetup -v –verify-passphrase luksFormat /dev/sdXY
- Ativar a criptografia do device
  - # cryptsetup luksOpen /dev/sdXY nome-virtual
  - [ou] # cryptsetup open --type luks /dev/sdXY nome-virtual

  Formatar a partição com algum sistema de arquivos
  - # mkfs.<fs> /dev/mapper/nome-virtual
- Montar a partição
  - # mount /dev/mapper/nome-vitual /ponto/montagem
- Para montagem durante o boot, configurar o arquivo /etc/crypttab
  - nome-virtual /dev/sdXY none luks
  - nome-virtual /dev/sdXY /arquivosenha luks
- Configurar o /etc/fstab
  - /dev/mapper/nome-virtual /ponto/montagem auto defaults 0 0

Outras opções:

- luksDump – Exibe informações do cabeçalho e informações de um dispositivo que está utilizando o LUKS
  - # cryptsetup luksDump /dev/sdXY

Tópico 204 – Administração Avançada de Dispositivos de Armazenamento

204.1 – Configurando RAID

RAID

Método para integrar vários dispositivos de armazenamento em um única unidade lógica. Pode ser feito diretamente no hardware ou via software (SO).

Tipos/Níveis de RAID (Cobrados pela LPI):

- RAID 0 (stripping) - Distribui os dados entre os discos, como se fosse um só. Sem redundância.
- RAID 1 (mirroring) - Espelha os dados nos discos. Os dois (ou mais) discos possuem os mesmos dados. Se um arquivo é apagado em um disco, também será apagado no outro. O RAID 1 protege os dados contra falhas do dispositivo.
- RAID 5 – Exige no mínimo 3 discos. A redundância é distribuída nos três discos através do uso de bits de paridade.

Tipo de Partição

Uma partição a ser utilizada em um dispositivo RAID deve ser definida com o tipo 0xFD, que define o “Linux RAID Autodetect”.

Criando RAID com o comando mdadm

A seguir as principais opções formas de uso do comando mdadm:

Criar um device RAID

mdadm -v --create /dev/md0 -l1 -n2 /dev/sdc1 /dev/sdd1

- -v ou --verbose : Exibe detalhes do processo
- -C ou --create : Cria um RAID
- -l ou --level : Define o tipo/nível do RAID (0, 1, 5, etc)
- -n ou --raid-devices : Define quantos membros participarão do RAID

Formatar o dispositivo RAID

Após criado, o dispositivo RAID deve ser formato com algum filesystem, por exemplo:

mkfs.ext4 /dev/md0

Configurar o mdadm.conf

mdadm --verbose --detail --scan >> /etc/mdadm/mdadm.conf
* ou /etc/mdadm.conf

Parar o dispositivo

mdadm --stop /dev/md0
Reiniciar o dispositivo

mdadm --assemble --run /dev/md0

Define uma partição/disco como “em falha”

mdadm --manage --fail /dev/md0 /dev/sdc1
* --fail ou -f ou --set-faulty

Verificar o status de um dispositivo RAID

mdadm --detail /dev/md0

Verificar o status de uma partição/disco que pertence a um dispositivo RAID

mdadm --examine /dev/sdc1

Apagar o superblock de uma partição/disco

mdadm --zero-superblock /dev/sdc1

Re-escanear os dispositivos e arquivo de configuração para iniciar os RAIDs:

mdadm --assemble --scan

O “arquivo” /proc/mdstat

O /proc/mdstat contém as informações dos dispositivos RAID ativos no momento. Exemplo:

cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md0 : **active raid5 sdc2[4] sdc1[0] sdd1[3]**
      20953088 blocks super 1.2 level 5, 512k chunk, algorithm 2 **[3/3] [UUU]**

Nesse caso o dispositivo md0 é um RAID5, com 3 device sendo que os 3 estão ativos e em funcionamento.

204.2 – Ajustando o Acesso aos Dispositivos de Armazenamento

Comando hdparm

Função: Visualizar e Alterar parâmetros de dispositivos IDE/ATA/SATA

Uso e Principais Opções:

hdparm /dev/hda – Exibir parâmetros do dispositivo
hdparm -i /dev/hda – Exibir informações relacionadas aos drivers do dispositivo
hdparm -I /dev/hda – Exibir informações detalhadas do dispositivo
hdparm -t /dev/hda – Realizar testes de leitura no device
hdparm -d1 /dev/hda – Definir o uso do DMA (Direct Memory Access) no dispositivo IDE/ATA

Comando sdparm

Função: Visualizar e Alterar parâmetros de dispositivos SCSI ou SATA

Uso e Principais Opções:

sdparm /dev/sda – Exibir parâmetros do dispositivo
sdparm -a /dev/sda – Exibir todos os parâmetros e modos de operação

* Dispositivos SATA não utilizam DMA

Comando nvme

Função: Listar e enviar operações a um controlador NVMe

Uso e Principais Opções:

nvme list – Listar dispositivos em uso através do NVMe

* O NVM-Express é usado para que discos SSD (Solid State Disk) possam ser utilizados através de barramentos PCI Express.

Comando fstrim

Função: O comando é usado em dispositivos SSD e em Storages (NAS) para descartar (“trim”) blocos que não estão em uso pelo filesystem, reduzindo problemas de performance por fragmentação.

Uso e Principais Opções:

fstrim /diretorio/montado
fstrim -a : Faz o trim em todos os diretórios montados que suportem o recurso.

* O dispositivo deve suportar o recurso de TRIM

Dispositivos IDE-ATA/SATA/SCSI/SSD

- IDE-ATA são mapeados como:
  - Primário
    - Master: /dev/hda
    - Slave: /dev/hdb
  - Secundário
    - Master: /dev/hdc
    - Slave: /dev/hdd
- SATA/SCSI, mapeados no Linux como: /dev/sd*
- Dispositivos que utilizam NVM-Express mapeados como: /dev/nvme*

SAN (Storage Area Network)

NAS = Network Attached Storage
SAN = Storage Area Network
FCP = Fibre Channel Protocol
FCoE = Fibre Channel over Ethernet
AoE = ATA over Ethernet

WWID = World Wide Identifier
WWN = World Wide Name

Ambos são identificadores muito utilizados em redes SAN para identificar os dispositivos de armazenamento de maneira única na rede.

Em um SAN Fibre Channel, o WWID/WWN tem a mesma função do endereço MAC address em uma rede Ethernet. No entanto, seu uso pode variar conforme o ambiente e tecnologias utilizadas.

LUN = Logical Unit Number

No protocolo SCSI, é utilizado para identificar uma Unidade Lógica de armazenamento, podendo ser um dispositivo único, uma parte ou até um grupo de dispositivos.

Comandos Úteis:

cat **/proc/scsi/scsi** : Exibe os dispositivos mapeados como SCSI no Linux, incluindo as informações dos dispositivos, do canal, e o LUN.

/lib/udev/**scsi_id** -g /dev/sda : Obter o WWID associado a um dispositivo no Linux

ls -l **/dev/disk/by-id** : Exibe os WWIDs associados a todos os dispositivos.

---

iSCSI (Internet Small Computer System Interface)

Protocolo que permite o compartilhamento de dispositivos de bloco via rede IP. Muito usado para conectar clientes a servidores NAS (Storage)

Clientes/Initiator veem o disco como locais, não como em rede.

Target (Servidor)

Utilizando o pacote tgt, a configuração é feita no arquivo: **/etc/tgt/targets.conf**:

<target iqn.2018-06.br.com.exemplo.server.target1>

backing-store /dev/sdb1

direct-store /dev/sdd

</target>

Initiator (Cliente)

iscsid – Daemon do Open-iSCSI que monitora e gerencia as conexões do iSCSI

Arquivo de configuração do iscsid: **/etc/iscsi/iscsid.conf**

Comandos:

- Descobrir os targets disponívieis:
  - # iscsiadm -m discovery -t sendtargets -p IP
- Conectar ao target:
  - # iscsiadm -m node -p IP --target=xxxxx --login
- Exibir informações de conexões estabelecidas:
  - # iscsiadm -m session -P3
- Desconectar dos targets:
  - # iscsiadm -m node -u

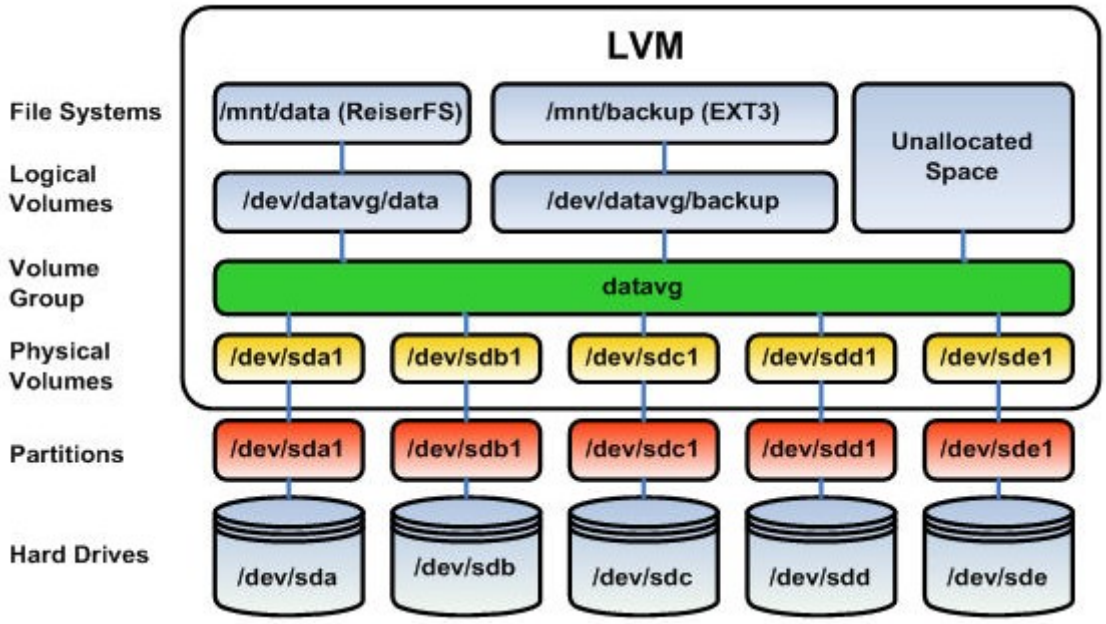
    - # iscsiadm -m node -U all
- Apagar todas as referências aos targets descobertos:
    - # iscsiadm -m discovery -p IP -o delete

Informações sobre os targets também podem ser encontradas no diretório **/var/lib/iscsi/**

204.3 – LVM – Logical Volume Manager

Cinco Elementos Fundamentais:

- PV: Physical Volume: O disco rígido ou uma partição do disco ou qualquer dispositivo de armazenamento
- VG: Volume Group: Um conjunto de PVs
- LV: Logical Volume: O equivalente a uma partição do disco tradicional, mas lógico.
- PE: Physical Extend: Faixas de dados de um volume físico. O padrão é 4 Mb.
- LE: Logical Extend: Faixas de dados de um volume lógico. Mesmo tamanho que o PE.



LVM2

Arquivo de Configuração do LVM: /etc/lvm.conf

Partições que integrarão o LVM devem ter o tipo “8e”.

Principais Comandos:

- **PV (Physical Volume)**
  - pvcreate : Cria um PV a partir de uma partição
    - # pvcreate /dev/sdb1
  - pvs / pvdisplay : Exibe informações dos PVs
  - pvremove : Remove um PV
  - pvscan : Busca todos os discos em busca de PVs
- **VG (Volume Group)**
  - vgcreate : Cria um VG a partir de um conjunto de PVs
    - # vgcreate grupo1 -sX /dev/sda1 /dev/sdb1 …
  - vgs / vgdisplay : Exibe informações dos VGs
  - vgchange : Muda propriedas do VG, usado por exemplo para ativar o VG
    - # vgchange -a y grupo1
  - vgextend : Aumentar o VG adicionando PVs
  - vgreduce : Reduzir o VG removendo PVs não utilizados
  - vgscan : Busca VGs por todos os discos e reconstrói os caches
  - vgrename : Renomeia o VG
  - vgremove : Remove um VG
- **LV (Logical Volume)**
  - lvcreate : Cria um LV a partir de um VG
    - # lvcreate -L10G grupo1 -n lv1
    - # lvcreate -L1G -s -n exemplo-snap /dev/grupo1/lv1 (criar um snapshot)
  - lvs / lvdisplay : Exibe informações dos LVs
  - lvextend : Aumenta o tamanho de um LV
  - lvreduce : Diminui o tamanho de um LV
  - lvrename : Renomeia um LV
  - lvremove : Remove um LV
  - lvscan : Busca em todos os discos em busca de LVs
- resize2fs : Redimensiona um FS ext*.

Tópico 205 – Configurações de Rede

205.1 – Configurações Básicas de Rede

Comando ifconfig

Uso e Principais Opções

ifconfig : Exibir apenas as interfaces ativas (UP/RUNNING)
ifconfig -a : Exibir todas as interfaces de rede, inclusive as inativas
ifconfig eth0 : Exibir as informações de uma interface específica

ifconfig eth0 192.168.1.10 up : Definir um IP e subir a interface
ifconfig eth0 192.168.1.10 netmask 255.255.255.0 : Definir IP e Máscara em uma interface
ifconfig eth0 down : baixar uma interface

ifconfig eth0:0 10.0.0.10 netmask 255.255.255.0 : Definir um IP adicional à uma interface
ifconfig eth0:sub1 172.16.30.1 : Definir um IP adicional à uma interface
ifconfig eth0 add 10.0.0.10 netmask 255.255.255.0 : Definir um IP adicional à uma interface

ifconfig eth0 hw ether 00:00:00:00:00:00 : Definir um MAC Address específico para uma interface

---

Comando route

Uso e Principais Opções

route -n : Exibe as rotas ativas, sem resolver os nomes

route {add|del} [-net|-host] alvo [interface]

route add default gw 192.168.1.1 : Define o IP gateway. Pode ser incluído o “dev eth0” ao final.
route del default : Remover a rota padrão.

route add -net 172.16.10.0 netmask 255.255.255.0 gw 192.168.1.10 dev eth0 : Adicionar uma rota à rede 172.16.10.0, pelo gateway/roteador 192.168.1.10, utilizando a interface eth0
route add -net 172.20.0.0/16 gw 192.168.1.1

route del -net 172.20.20.0/24
route del -net 10.0.0.0 netmask 255.255.255.0 : Remover a rota

---

Comando ip

Uso e Principais Opções

ip link show : Exibe o estado de cada interface
ip link show eth0 : Exibe o estado de uma interface específica

ip address show : Exibe as interfaces com seus respectivos IPs
ip -6 address show : Exibe apenas os IPs IPV6 de cada interface
ip address flush dev eth0 : Limpa as definições de IP
ip address add 10.0.0.10/24 dev eth0 : Define um novo IP à interface

ip route show : Exibe as rotas
ip route add 172.16.10.0/24 via 192.168.1.50 : Adiciona uma rota
ip route del 172.30.30.0/24
ip route add default via 192.168.1.1
ip route del default

ip neigh show : Exibe a tabela ARP (IP x MAC Address)

ARP = Address Resolution Protocol

Protocolo utilizado para resolver endereços da camada de Internet (IP) em endereços da camada de enlace (MAC).

Comando arp

Função: Exibir e manipular a tabela de registros ARP

Uso e Principais Opções:

arp : Exibe a tabela IP x MAC atual
arp -n : Não resolve nomes
arp -d 192.168.1.10 : Remove a referência da tabela
arp -s 192.168.1.10 00:00:00:00:00:00 : Adiciona uma referência específica manualmente
arp -f /arquivo : Adiciona manualmente as referências contidas no arquivo especificado. Se nenhum arquivo for informado, será utilizado o /etc/ethers

Comando arpwatch

Função: Monitorar e enviar alarmes ao administrador de sistema sobre mudanças nas referências da tabela ARP.

Uso e Principais Opções

arpwatch -d : Modo debug
arpwatch -e email@email : Envia alarmes a um e-mail específico
arpwatch -i eth0 : Especifica a interface a ser monitorada

Configurações de Redes Wireless

Comando iw

Função: Usado para configurar os dispositivos wireless. Suporta apenas o padrão 802.11.

Uso e Principais Opções:

iw dev wlan0 link : Exibir os status da interface e conexão wireless
iw dev wlan0 scan : Exibir as rede acessíveis pela interface
iw dev wlan0 connect "Access Point" 2432 : Estabelecer uma conexão a um SSID

Comando iwlist

Função: Scanear e listar as redes wireless disponíveis

Uso e Principais Opções:

iwlist wlp2s0 scanning

Comando iwconfig

Função: Configurar uma interface de rede wireless. Associar uma interface a uma rede wireless Equivalente ao ifconfig.

Uso e Principais Opções:

iwconfig essid “Rede Wireless X”
iwconfig key s:123456

205.2 – Configurações e Resolução de Problemas Avançados em Redes

Roteamento

Identificando Flags nas Tabelas de Rotas

- U: Rota ativa (UP)
- H: Alvo da rota é um host
- G: Gateway sendo utilizado
- R: Rota Reestabelecida por Roteamento Dinâmico
- D: Rota Instalada por Roteamento Dinâmico
- M: Rota Modificado por Roteamento Dinâmico
- ! : Rota Rejeitada

Alguns Daemons de Roteamento Dinâmico:

- **Routed**
- **GateD**
- BIRD
- Zebra
- Quagga

Para habilitar o roteamento de pacotes de uma rede para outra, o seguinte parâmetro deve ser configurado para 1 no kernel:

- /proc/sys/net/ipv4/ip_forward (IPv4)
- /proc/sys/net/ipv6/conf/all/forwarding (IPv6)

Testando o Acesso a Hosts

Comandos ping / ping6

O comando ping pode ser utilizado tanto para IPv4 quanto para IPv6, com o uso da opção -6. O ping6 é específico para IPv6.

O ping utiliza o protocolo **ICMP**, enviando um pacote de **ECHO_REQUEST** e aguardando o retorno do **ECHO_REPLY**

Comandos traceroute / traceroute6

Assim como o ping, o traceroute pode ser utilizado tanto para IPv4 quanto para IPv6, com o uso da opção -6. O traceroute6 é específico para IPv6.

O traceroute utiliza por padrão o **protocolo UDP** para realização dos testes. Utilizando a opção -I o ICMP será utilizado.

O comando também utiliza o campo **TTL (Time to Live)** do protocolo IP para testar as conexões. Caso um host não consiga alcançar o host seguinte, ele deve retornar um pacote **ICMP TIME_EXCEEDED**.

Comando netcat (nc)

Função: Utilizado para testar conexões em portas TCP e UDP

Uso e Principais Opções:

nc 192.168.1.10 80 : Testar a porta 80 no IP 192.168.1.10
nc -vz 192.168.1.10 50-100 : Testa as portas de 50 a 100 do IP informado. O -z faz com que a porta seja apenas verificada mas não é feita a conexão. O -v é o modo verbose.

nc -l -p 1234 : Abre no host local o processo para escuta (listenning) na porta 1234

Comando netstat

Função: Exibir as conexões de rede da máquina, tabelas de roteamento, estatísticas sobre as interfaces e etc.

Uso e Principais Opções:

- -r : Exibe a tabela de rotas
- -n : Não resolve nomes
- -a : Exibe tanto as conexões de sockets em listenning quanto as demais. Sem o -a (ou o -l), as conexões em listenning não são exibidas.
- -l : Exibe os sockets em estado listening
- -i : Exibe estatísticas por interface de rede
- -t : TCP
- -u : UDP
- -A inet/unix/etc : Especificar o tipo de conexão. -A inet = --inet
- -p : Exibe o PID/Processo responsável por cada socket
- -c : Atualiza continuamente
- -s : Exibe as estatísticas por protocolo

Comando ss

Função: É o comando que visa substituir o netstat. Possui as mesmas funções de investigação e análise de sockets e conexões abertas.

Uso e Principais Opções:

Muitas das opções são iguais ao netstat, com algumas exceções:

- -a : Exibe todos os sockets
- -n : Não resolve os nomes dos serviços

- -r : Resolve os nomes dos hosts. No ss, por padrão os nomes dos hosts não são traduzidos
- -l : Conexões em listenning
- -p : Exibe os processos
- -t : TCP
- -u UDP
-

Comando lsof

Função: Listar arquivos abertos. Como todo socket também possui um arquivo aberto no sistema, ele também pode ser usado para monitorar conexões.

Uso e Principais Opções:

lsof -i : Exibe os arquivos relacionados ao protocolo IP
lsof -i tcp : Apenas TCP
lsof -i :443 : Apenas porta 443
lsof -i @192.168.1.1:443 : Apenas de um IP específico em uma Porta específica

---

Comando tcpdump

Função: Realizar uma análise dos pacotes que trafegam pelas interfaces de rede.

Uso e Principais Opções:

- -c : Limita a quantidade de pacotes a ser capturados
- -D : Apenas lista as interfaces que podem ser analisadas
- -i : Especifica uma interface para ser capturada
- -q : Exibe os resultados em modo sucinto
- -t : Não mostra hora
- -v -vv -vvv : Ativa diferentes níveis de detalhamento
- -w : Escreve o resultado em um arquivo
- -r : Lê os dados de um arquivo

Filtros:
- dst host : Especificar o host destino
- src host : Especificar o host origem
- host : Especificar um host para origem ou destino
- dst port : Especificar uma porta destino
- src port : Especificar uma porta origem
- port : Especificar uma porta para origem ou destino
- dst net : Especificar uma rede destino
- src net : Especificar uma rede origem
- net : Especificar uma rede para origem ou destino
- dst portrange : Especificar um grupo de portas destino
- src portrange : Especificar um grupo de portas origem
- portrange : Especificar um grupo de portas para origem ou destino

Os filtros podem ser combinados com “and” e “or”. O not ou ! também pode ser uado para negar uma das opções.

Comando nmap

Função: É um port scanner, ou seja, uma ferramenta capaz de fazer uma varredura de portas locais ou de hosts e redes remotas.

Uso e Principais Opções:

nmap 192.168.1.10 : Faz a varredura em um host específico
nmap 192.168.1.10/24 : Faz a varredura em todos os hosts de uma rede

Opções:

- -F : FastScan. Verifica um conjunto menor de portas
- -sV : Procura identificar mais detalhes relacionados à versão do serviço disponibilizado em cada porta
- -p : Analisa uma porta específica, ou um conjunto delas.
- -O : Identifica detalhes do Sistema Operacional através do Stack Fingerprint

205.3 – Resolução de Problemas em Redes

Configurações de Rede

Padrão Debian

Arquivo de Configuração das Interfaces e Gateway: **/etc/network/interfaces**

Exemplo:
auto enp0s3
iface enp0s3 inet static
    address 192.168.1.210
    netmask 255.255.255.0
    gateway 192.168.1.1

Padrão RedHat

Arquivo de Configuração das Interfaces: /etc/sysconfig/network-scripts/ifcfg-*

Exemplo: ifcfg-enp0s3
DEVICE=enp0s3
BOOTPROTO=static
IDPADDR=192.168.1.210
NETMASK=255.255.255.0
ONBOOT=yes

O gateway deve ser configurado no arquivo: /etc/sysconfig/network:
GATEWAY=192.168.1.1

NetworkManager

Função: Serviço responsável por gerenciar de maneira dinâmica as interfaces de rede com ou sem fio e suas conexões com a rede.

Diretório Principal: /etc/NetworkManager/

Comando Principal: nmcli

Investigação de Eventos e Logs

No processo de investigação de falhas de redes, informações e registros devem ser analisados principalmente nos seguintes arquivos e comandos:

- dmesg
- /var/log/messages (RedHat)
- /var/log/syslog (Debian)
- Systemd Journal, pelo comando “journalctl”

Falhas de DNS

Em eventos de falhas de resolução de nomes, os seguintes arquivos de configuração devem ser analisados:

- /etc/resolv.conf - Configurações referentes ao serviço DNS, principalmente o(s) servidor(es) DNS que deverá(ão) ser utilizado(s), através do parâmetro "nameserver"
- /etc/hosts - Definições estáticas de IP e Nomes
- /etc/networks - Definições estáticas de Redes e Nomes
- /etc/hostname ou /etc/HOSTNAME - Nome da máquina

Comandos para análise de DNS:

- hostname : Nome da máquina
  - -d : Exibe o domínio
  - -f : Exibe o FQDN (Fully Qualified Domain Name), ou seja, o nome completo do host
- host : Ferramenta utilizada para verificar a resolução de nomes
  - -a : Verifica todos os registros relacionados a um nome
  - -t <tipo> : Verifica um tipo de registros específico
- dig : Ferramenta para realizar a verificação de DNS de um endereço
  - # dig www.lpi.org
  - # dig www.lpi.org @8.8.8.8 : Utiliza um servidor DNS específico
  - # dig lpi.org -t mx : Verifica apenas um tipo específico de registro, no caso MX

TCP Wrappers

Os arquivos de configuração /etc/hosts.allow e /etc/hosts.deny podem ser configurados para definir quem pode ou não utilizar serviços que utilizam as bibliotecas do TCP Wrappers

É importante notar que:

- Se não há nenhuma configuração em nenhum dos 2 arquivos, não há bloqueios
- Se há uma regra de liberação de um serviço para determinado IP no /etc/hosts.allow, o host.deny nem mesmo é consultado
- Se não há regras de liberação no hosts.allow, o hosts.deny é consultado

Exemplos para o /etc/hosts.allow:
ALL: 192.168.8.* EXCEPT 192.168.8.1 : Libera todos os serviços para a rede 192.168.8.0/24, exceto para o IP 192.168.8.1
telnetd: 192.168.8.10 192.168.8.50 : Libera o serviço telnet para os 2 IPs informados

Exemplos para o /etc/hosts.deny:
ALL: ALL : Bloqueia tudo o que não for liberado no hosts.allow
sshd: 192.168.1.* : Bloqueia o serviço SSH para a rede 192.168.1.0/24

Comando mtr

Função: O mtr (My Trace Route) combina as funções do ping com o traceroute em uma interface que faz a atualização constante das informações de acesso a um host específco.

Uso e Principais Opções:

mtr -n www.lpi.org

Tópico 206 – Manutenção do Sistema

206.1 – Compilar e Instalar Programas pelo Código Fonte

Compressão

Comandos usados para comprimir e descomprimir:

- Gzip = gzip / gunzip / tar z
- Bzip2 = bzip2 / bunzip2 / tar j
- xz = xz / unxz / tar J

Para realizar o processo de compilação de um software, ele normalmente é descompactado no diretório **/usr/src**, que é por padrão o diretório de fontes do sistema.

Comando/Script configure

Função: O primeiro passo, após a descompactação da fonte, é utilizar o script **configure**, que é disponibilizado junto com a maioria dos códigos fonte, para configurar o arquivo "Makefile", que será utilizado durante o processo de compilação.

O arquivo "**Makefile.in**" serve como base para o "**configure**" que então irá gerar o "**Makefile**".

Uso e Principais Opções:

./configure

Uma opção comum de ser utilizada como o configure é o --prefix, que indica em qual diretório a aplicação será instalada no final do processo.

./configure --prefix=/opt/software/

Comando make

Função: O comando **"make"** fará a compilação do código fonte, utilizando como base o arquivo Makefile, gerando no final do processo o binário executável do software.

Uso e Principais Opções:

make

Comando make install

Função: O comando **“make install”** copiará os binários, bibliotecas, arquivos de configuração e afins para os diretórios definidos para o software, configurado no Makefile.

Uso e Principais Opções:

make install

Comando patch

Função: Aplica as diferenças definidas em um arquivo de patch ao(s) arquivo(s) de código fonte presentes, da versão anterior.

Um arquivo de patch é um diff entre dois arquivos ou entre dois diretórios, normalmente entre o diretório de código fonte de uma nova versão com a versão anterior.

Uso e Principais Opções:

Aplicando um patch:

patch < arquivo.patch

Revertendo um patch

patch -R < arquivo.patch

Opções:

- -p<num>, ex -p0, -p1, -p2… : Remove <num> / do nome do caminho do arquivo. -p0 não modifica em nada. -p1 remove apenas a barra inicial.

206.2 – Operações de Backup

Mídias para Backup

- **Discos**
  - Vantagens:
    - Velocidade no Acesso (Localização da Informação)
    - Facilidade de Gerenciamento
    - Sempre pronto para uso
  - Desvantagens:
    - Menor vida útil
    - Mais frágil
- **Fitas**
  - Vantagens
    - Durabilidade (pode durar até 30 anos)
    - Compacta / Facilidade de Transporte
    - Baixo custo por GB
    - Grande Capacidade
  - Desvantagens
    - Gravação e Acesso Sequencial
    - Na necessidade de uso, a fita correta deve ser localizada
- CD/DVD/Blu-Ray
- Pendrives/Cartões de Memória

Tipos de Backup

- **Backup Full** - Backup Completo
  - Vantagens:
    - Processo de geração e recuperação mais simples
  - Desvantagens:
    - Maior demora na geração de cada backup
    - Ocupa mais espaço
- **Backup Incremental** - Após um backup full, os backups seguintes copiam apenas os dados alterados desde o backup imediatamente anterior
  - Vantagens:
    - Backups são gerados mais rapidamente
    - Ocupa menos espaço em disco
  - Desvantagens:
    - Necessidade de maior controle e organização
    - Processo de recuperação mais lento e trabalhoso. Todos os backups desde o full devem ser recuperados.
- **Backup Diferencial -** Após um backup full, os backups seguintes copiam apenas os dados alterados a partir do backup full.
  - Vantagens:
    - Recuperação mais simples. Apenas o backup full e um backup diferencial precisam ser recuperados.

- Desvantagens:
  - Exige maior volume de dados se comparados com o backup incremental

---

Soluções de Backup Open Source

- Bacula
- Amanda
- Bareos (fork do Bacula)
- BackupPC

Diretórios para Backup

Diretórios normalmente incluídos nos processos de backup: /home – Arquivos e Diretórios dos Usuários

- /etc - Arquivos de configuração do S.O. e das aplicações e serviços
- /var
  - /var/log - Arquivos de log
  - /var/db, /var/lib - Banco de Dados
  - /var/spool/mail - E-mails
  - /var/www - Páginas web hospedadas no servidor
- /usr e /opt - Caso se deseje um backup das aplicações instaladas.

Comando tar

Função: Realizar atividades de arquivamento, agrupamento e compactação de conjuntos de arquivos.

Uso e Principais Opções:

- -c --create : Criar um novo agrupamento de arquivos
- -x --extract : Extrair arquivos
- -t --list : Listar o conteúdo de um arquivo
- --exclude : Exluir um arquivo ou diretório do tar
- -g --listed-incremental : Fazer um backup incremental

```
# tar -cf aquivo.tar /etc
# tar -c /etc > arquivo.tar

# tar -cf /dev/st0 /etc      : Criar o backup em um dispositivo de fita
```

tar -xf /dev/nst0 : Extrair os dados de uma fita

tar -cf backup.tar -g controle.txt /etc : Criar um arquivo para backup incremental

---

Comando cpio

Função: Criar/Extrair arquivos agrupados no formato cpio.

Uso e Principais Opções:

find /etc/*.conf | cpio -o > backup.cpio : Gerar um arquivo

cpio -ivdm < backup.cpio : Extrair um arquivo completo

cpio -ivdm < backup.cpio '*.bin' : Extrair apenas alguns arquivos do .cpio

find /etc/*.conf | cpio -o | gzip -c > backup.cpio : Gerar um arquivo comprimido

---

Comando dd

Função: Fazer uma cópia completa de ou para uma partição.

Uso e Principais Opções:

dd if=/dev/sda1 of=backup.img

dd if=backup.img of=/dev/sda1

Comando scp/ssh

Função: Enviar ou obter arquivos de forma segura entre hosts.

Uso e Principais Opções:

scp arquivo.tgz usuario@host:/diretorio : Enviando um arquivo

scp usuario@host:/diretorio/arquivo.tgz . : Obtendo um arquivo

---

Comando rsync

Função: Copiar arquivos entre diretórios de uma mesma máquina ou remotamente, mantendo os arquivos/diretórios em sincronia.

Uso e Principais Opções:

- -a : Arquive mode. Inclui diversas opções, entre elas faz a cópia recursiva, mantém permissões e data/hora.
- -r : Copia recursiva de diretórios
- -u : Copia apenas arquivos modificados desde a última execução
- -z : Comprime os arquivos durante o processo de transferência
- -p : Mantém as permissões
- -v : Modo de detalhamento
- --progress : Mostra o progresso durante as transferências
- --delete : Remove os arquivos no destino caso eles não existam mais na origem
- -e ssh: Faz a conexão segura via SSH

rsysnc -auv --delete /var/log usuario@host-remoto:/opt/backup/

rsync -ru -e ssh /var/log usuario@host-remoto:/opt/backup/

Comando mt

Função: Gerenciar dispositivos de fita

Uso e Principais Opções:

Os dispositivos de fita são mapeados no Linux como **/dev/st0 e /dev/nst0.** Os dispositivos /dev/st0 são rebobinados automaticamente após uma operação, os nts0 precisam ser rebobinados via comando.

mt -f /dev/st0 <opção>

- rewind : rebobina a fita
- offline : rebobina e descarrega a fita do dispositivo
- status : mostra um status da fita
- erase : apaga o conteúdo da fita

206.3 – Notificar os Usuários do Sistema

Comando wall

Função: Enviar uma mensagem a todos os usuários logados no sistema

Uso e Principais Opções:

wall mensagem

echo “mensagem” | wall

cat arquivo | wall

Comando shutdown

Função: Enviar comandos para que o sistema seja reiniciado ou desligado, possibilitando o agendamento e envio de mensagens personalizadas aos usuários conectados.

Uso e Principais Opções:

- -H --halt : “Desliga” o sistema operacional.
- -P --poweroff : “Desliga” o sistema operacional e a energia da máquina (se suportado)
- -r --reboot : Reinicializa o sistema
- -c : Cancela a operação programada anteriormente
- --no-wall : Não envia nenhuma mensagem aos usuários

shutdown -r 18:00 “O sistema será reinicializado”

Arquivos de Configuração de Mensagens

- /etc/issue – Mensagem exibida antes de um login local
- /etc/issue.net – Mensagem exibida antes de um login remoto
- /etc/motd - “Message of the day”. Mensagem exibida após o login.