Red Hat System Administration II

After The Course I’ll Be Able To:

- Attach and configure a workstation on an existing network
- Pass (isa :)) Red Hat Certified Administration exam.

Course Outlines

- Installation
- System Initialization and Services
- Tuning and Maintaining the kernel
- Managing System Logs
- Filesystem Management
- Network Configuration and Troubleshooting
- Installing and managing Software
- Users and Groups Administration
- Command line Process Management
- Printing and Administration Tools
- Advanced Filesystem Management

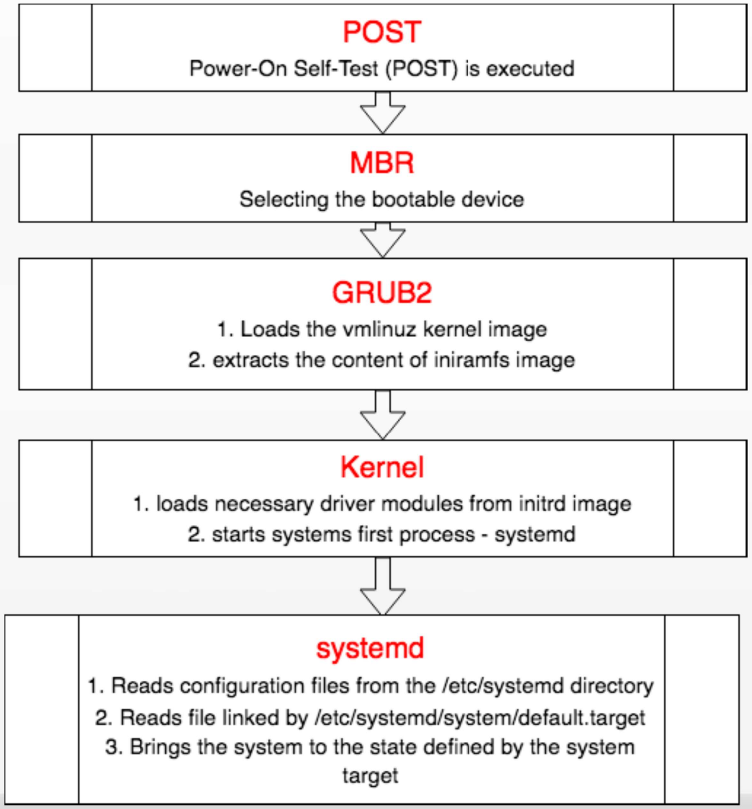
What about Today?

- Explain System initialization steps and related processes

Boot Sequence

Introduction

- It is important to understand the Linux boot process to troubleshoot boot problems.



POST

Power-On Self-Test (POST) is executed

MBR

Selecting the bootable device

GRUB2

1. Loads the vmlinuz kernel image
2. extracts the content of iniramfs image

Kernel

1. loads necessary driver modules from initrd image
2. starts systems first process - systemd

systemd

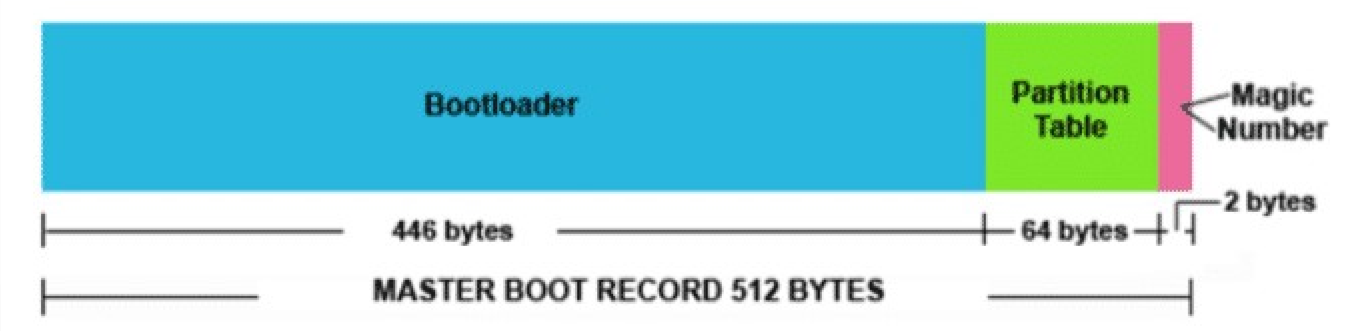
1. Reads configuration files from the /etc/systemd directory
2. Reads file linked by /etc/systemd/system/default.target
3. Brings the system to the state defined by the system target

POST
(Power on Self Test)

- From the system firmware, which can be
  - The modern Universal Extended Firmware Interface (**UEFI**)

    or
  - The classical Basic Input Output System (**BIOS**)
- The Power-On Self-Test (POST) is executed, and the system hardware is detected, tested and initialized and MBR is loaded (Master Boot Record).

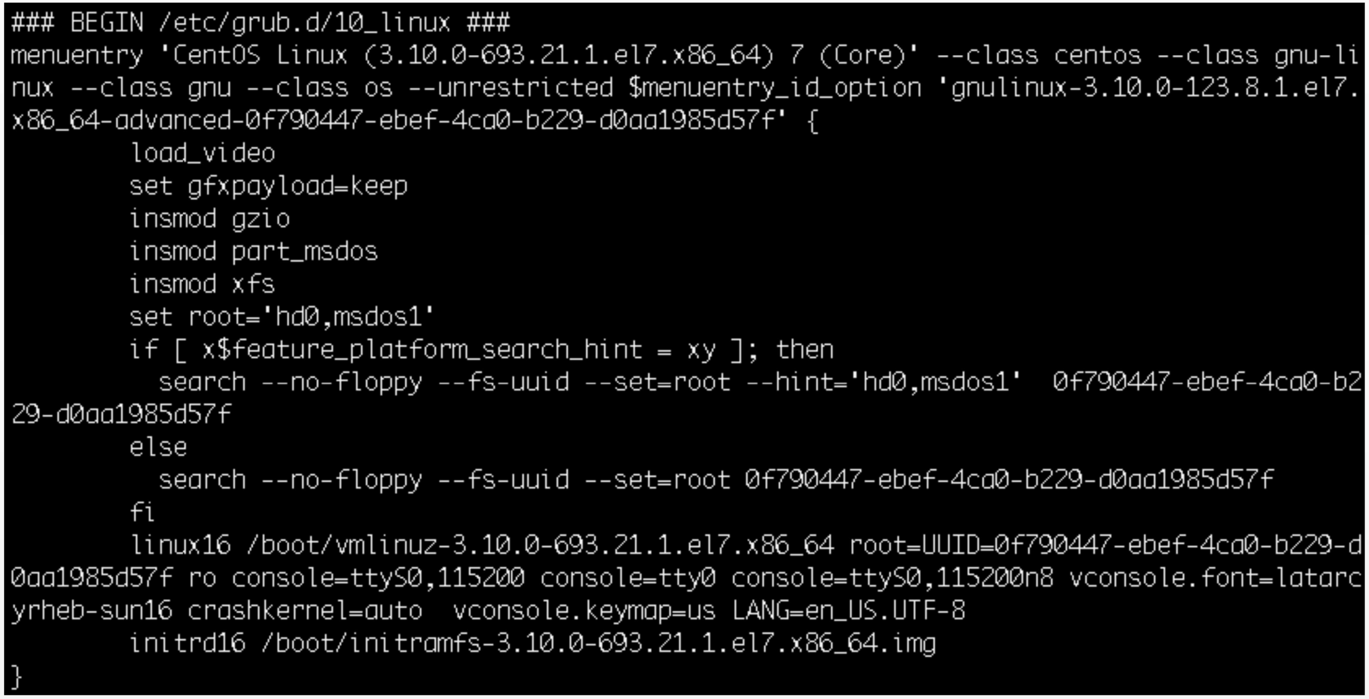
Master boot record (MBR)



- Master Boot Record (MBR) is the first 512 bytes of the boot drive that is read into memory by the BIOS.
- The next 64 bytes contain the partition table for the disk.
- The last two bytes are the “Magic Number” which is used for error detection.
- MBR discovers the bootable device and loads the GRUB2 boot loader into memory and transfers control over to it.

GRUB 2 Boot Loader

- The default bootloader program used on RHEL 7 is GRUB 2.
- GRUB stands for GRand Unified Bootloader.
  - The GRUB 2 configuration file is located at /boot/grub2/grub.cfg (Do not edit this file directly).



```
### BEGIN /etc/grub.d/10_linux ###
menuentry 'CentOS Linux (3.10.0-693.21.1.el7.x86_64) 7 (Core)' --class centos --class gnu-li
nux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-123.8.1.el7.
x86_64-advanced-0f790447-ebef-4ca0-b229-d0aa1985d57f' {
        load_video
        set gfxpayload=keep
        insmod gzio
        insmod part_msdos
        insmod xfs
        set root='hd0,msdos1'
        if [ x$feature_platform_search_hint = xy ]; then
          search --no-floppy --fs-uuid --set=root --hint='hd0,msdos1'  0f790447-ebef-4ca0-b2
29-d0aa1985d57f
        else
          search --no-floppy --fs-uuid --set=root 0f790447-ebef-4ca0-b229-d0aa1985d57f
        fi
        linux16 /boot/vmlinuz-3.10.0-693.21.1.el7.x86_64 root=UUID=0f790447-ebef-4ca0-b229-d
0aa1985d57f ro console=ttyS0,115200 console=tty0 console=ttyS0,115200n8 vconsole.font=latarc
yrheb-sun16 crashkernel=auto  vconsole.keymap=us LANG=en_US.UTF-8
        initrd16 /boot/initramfs-3.10.0-693.21.1.el7.x86_64.img
}
```

GRUB 2 Boot Loader (cont.)

- GRUB 2 menu-configuration settings are taken from /etc/default/grub when generating grub.cfg
- Sample /etc/default/grub file :

```
# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console”
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap
crashkernel=auto rd.lvm.lv=rhel/root rhgb quiet net.ifnames=0"
      GRUB_DISABLE_RECOVERY="true"
```

GRUB 2 Boot Loader (cont.)

- If changes are made to any of these parameters, you need to run **grub2-mkconfig** to re-generate the /boot/grub2/grub.cfg file.

  # grub2-mkconfig –o /boot/grub2/grub.cfg
- GRUB2 searches the compressed kernel image file also called as *vmlinuz* in the /boot directory.
- GRUB2 loads the vmlinuz kernel image file into memory and extracts the contents of the initramfs image file into a temporary, memory-based file system (tmpfs).
- The initial RAM disk (initrd) is an initial root file system that is mounted before the real root file system.

GRUB 2 Boot Loader (cont.)

- **initramfs**
  - The job of the initial RAM file system is to preload the block device modules, such as for IDE, SCSI, or RAID, so that the root file system, on which those modules normally reside, can then be accessed and mounted.
  - The **dracut** utility creates initramfs whenever a new kernel is installed.
  - Use the **lsinitrd** command to view the contents of the image created by dracut:

```
# lsinitrd | more
```

Kernel

- The kernel starts the systemd process with a process ID of 1 (PID 1)
- It also loads the necessary driver modules from initrd image.
- To load Linux, the kernel is loaded together with the initramfs. The initramfs contains kernel modules for all hardware that is required to boot, as well as the initial scripts required to proceed to the next stage of booting.
  - On RHEL 7, the initramfs contains a complete operational system (which may be used for troubleshooting purposes).

systemd

- systemd is the parent of all processes on a system.
- systemd is the first process that starts after the system boots, and is the final process that is running when the system shuts down.
- It controls the final stages of booting and prepares the system for use. It also speeds up booting by loading services concurrently.

systemd (cont.)

- It reads the file linked by /etc/systemd/system/default.target (for example, /usr/lib/systemd/system/multi-user.target) to determine the default system target (equivalent to run level).
- The system target file defines the services that systemd starts.

systemd (cont.)

- **Comparison of SysV Run Levels and Target Units**

Run Level	Target Units	Description
0	runlevel0.target, poweroff.target	Shut down and power off
1	runlevel1.target, rescue.target	Set up a rescue shell
2,3,4	runlevel[234].target, multi-user.target	runlevel[234].target, multi-user.target
5	runlevel5.target, graphical.target	Set up a graphical multi-user shell
6	runlevel6.target, reboot.target	Shut down and reboot the system

systemd (cont.)

- systemd brings the system to the state defined by the system target, performing system initialization tasks such as:
    1. Setting the host name
    2. Initializing the network
    3. Initializing SELinux based on its configuration
    4. Printing a welcome banner
    5. Initializing the system hardware based on kernel boot arguments
    6. Mounting the file systems, including virtual file systems such as the /proc file system
    7. Cleaning up directories in /var
    8. Starting swapping

systemd target units

- To view which target unit is used by default:

```
# systemctl get-default
        graphical.target
# runlevel
        N 5
```

- Notice that the default.target symbolic file points to graphical.target file.

```
# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 40 Oct 11 02:02 /etc/systemd/system/default.target->/usr/lib/systemd/system/graphical.target
```

systemd target units (cont.d)

- To change the default target unit to the graphical.target unit

```
# systemctl set-default graphical.target
```

```
rm '/etc/systemd/system/default.target‘
In -s '/usr/lib/systemd/system/graphical.target‘ '/etc/systemd/system/default.target‘
```

- To change currently active target

```
# systemctl isolate multi-user.target
```

Summary

Boot Phase	Configuration
POST	Hardware Configuration (F2, ESC, F10 or another key)
Select bootable Device	BIOS/UEFI configuration or hardware boot menu
Loading the boot loader	grub2-install and edits to /etc/defaults/grub
Loading the kernel	Edits to the GRUB configuration and /etc/dracut.conf
Switch to the root filesystem	/etc/fstab
Running the default target	/etc/systemd/system/default.target

Services

To start and stop services

- Start a service

```
# systemctl start service_name

# systemctl start network
```

- Stop a service

```
# systemctl stop service_name


# systemctl stop network
```

To check status of a service

```
# systemctl status service_name
# systemctl status network
```

To enabling or disabling (auto-start) of a service

- To enable auto-start a service at boot

```
#systemctl enable service_name

#systemctl enable network
```

- To disable auto-start a service at boot

```
#systemctl disable service_name

#systemctl disable network
```

Listing services

- List (enabled/disabled) status of all installed service units

```
# systemctl list-unit-files --type service
```

Print Tools

Introduction

- CUPS, the Common UNIX Printing System, is the primary printing system in Red Hat Enterprise Linux.
- Users can
  - submit jobs with lpr or lp
  - examine status with lpq or lpstat
  - abort their jobs with lprm or cancel

Identify user printer

- Users may specify a printer with an appropriate command-line option
  - If no printer is specified, both the lp and lpr commands look for a printer name by searching for a name in the following order
    - The environment variable PRINTER
    - The environment variable LPDEST
    - The first entry in /etc/printcap

Examples

```
lpr -P p1@off205 myfile
```

```
lp -d p1@off205 myfile
```

Configure printers using CUPS

- The lpadmin command
  - This text-only command allows you to
    - add printers
    - modify printers
    - delete printers

Configure printers using CUPS (cont.)

- The CUPS Web interface
  - Running on port 631
- GUI-based environment
  - system-config-printer utility

The configuration files

- The aforementioned commands modify the configuration files on your behalf
  - The /etc/cups/cupsd.conffile contains configuration data about the CUPS daemon.
  - The /etc/cups/printers.conf file contains configuration data about the printers.

Printer administration commands

- To allows print jobs to go to the printer queue
  - accept command
- To disallows print jobs from going to the printer queue
  - reject command
- To Allows print jobs from going from the print queue to the printer.
  - /usr/bin/enable command
- To disallows print jobs form going from the print queue to the printer
  - disable command

Scheduling and Processes

Task automation

- Red Hat Enterprise Linux provides two utilities to manipulate and schedule tasks
  - The at service
    - Executes a task at a specific time
    - Uses the at command to set up the execution of a task
    - Is administered by the atd daemon
  - The cron service
    - Executes tasks at scheduled intervals.
    - Uses the crontab command to set up the execution of a task.
    - Is administered by the crond daemon

Access to the at command

- /etc/at.allow and /etc/at.deny files restrict or allow users to run the at command.
- If /etc/at.allow exists and your username appears in it, you may use at command.
- If /etc/at.allow does not exist and /etc/at.deny does, then you must not be listed in /etc/at.deny to use at.
- If neither file exists, all users are denied to schedule jobs with at.
- The default installation supplies an empty at.deny file, thereby allowing the use of at to all users.

Use cron for task automation

- Crontab files are stored in /var/spool/cron
  - Avoid modifying these files directly, use crontab –e command
  - The cron file format

    Min(0-59) Hours(0-23)Day(1-31) Month(1-12)Day(0-6) Command
  - Example

    30 6 * * 0 /home/aya/mys1.sh

The /etc/anacrontab File

- It automates the execution of system tasks. These tasks are often vital for the integrity of the operating system

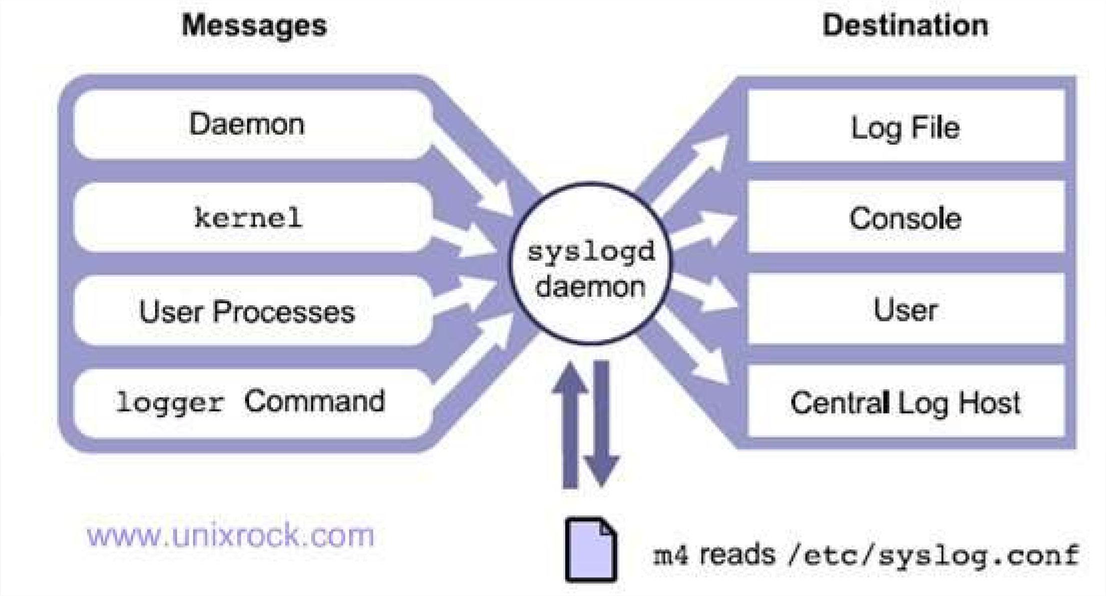
```
# cat /etc/anacrontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/bin:/usr/sbin
MAILTO=root
HOME=/
#period in days      delay in minutes      job identifier      command
   1                  5                     cron.daily          nice run-parts /etc/cron.daily
   7                  25                    cron.weekly         nice run-parts /etc/cron.weekly
@monthly              45                    cron.monthly                  nice run-parts
   /etc/cron.monthly
```

Managing System Logs

Introduction

- The rsyslog system messaging features track system activities and events.
- Log messages can be manually generated using logger command

How rsyslogd Works ?



rsyslogd Configuration File

- /etc/rsyslog.conf file
  - Selector
    - Facility: category of system processes that generate the messages
      - Example: kern, mail, daemon, cron, local0-7, *, ...
    - Level: severity or importance of the message
      - Example: emerg, alert, crit, err, warning,...
  - Action: where to send a message
    - Example: /pathname, @hostname, username, *
      - /pathname must already be exist

Monitoring a rsyslog File in Real Time

- You can monitor the designated rsyslog file in real time using the command tail -f.
  - The tail -f command holds the file open so that you can view messages being written to the file by the syslogd daemon.

```
tail -f /var/log/messages
jun 14 13:15:39 host1 inetd[2359]:[ID 317013 daemon.notice] telnet [ 2361] from 192.9.200.1 45800
Time/Date localhostname Pname[pid] MSG ID facility.level incoming request [ppid] IP Port name
```

Adding One-Line Entries to a System Log File

- The logger command enables you to send messages to the rsyslogd daemon.
  - A system administrator can write administrative shell scripts that report the status of backups, or other functions by using the logger command.

Adding One-Line Entries to a System Log File (cont.)

```
logger [ -i ] [ -f file ] [ -p priority ] [ -t tag ] [ message ]
```

Option	Description
-i	Logs the process Id of the logger command with each line
-f filename	Uses the content of the file as the message body
-p priority	Enters the message with the specified priority, the default is user.notice

Identify system log files

- The /var/log/dmesg file
  - Kernel log messages.
  - This log file is written after system boot.
  - It contains messages from the kernel that were generated during the boot process.
- The /var/log/messages file
  - It the standard system log file.
  - It contains messages from most of your system's services.

Identify system log files (cont.)

- The /var/log/maillog file
  - Mail system messages
  - It contains messages and errors from your mail service
- The /var/log/secure file
  - It contains messages and errors from security-related services such as login, network connections (xinetd)
  - Very useful in detecting and investigating attempts to gain unauthorized access to the system

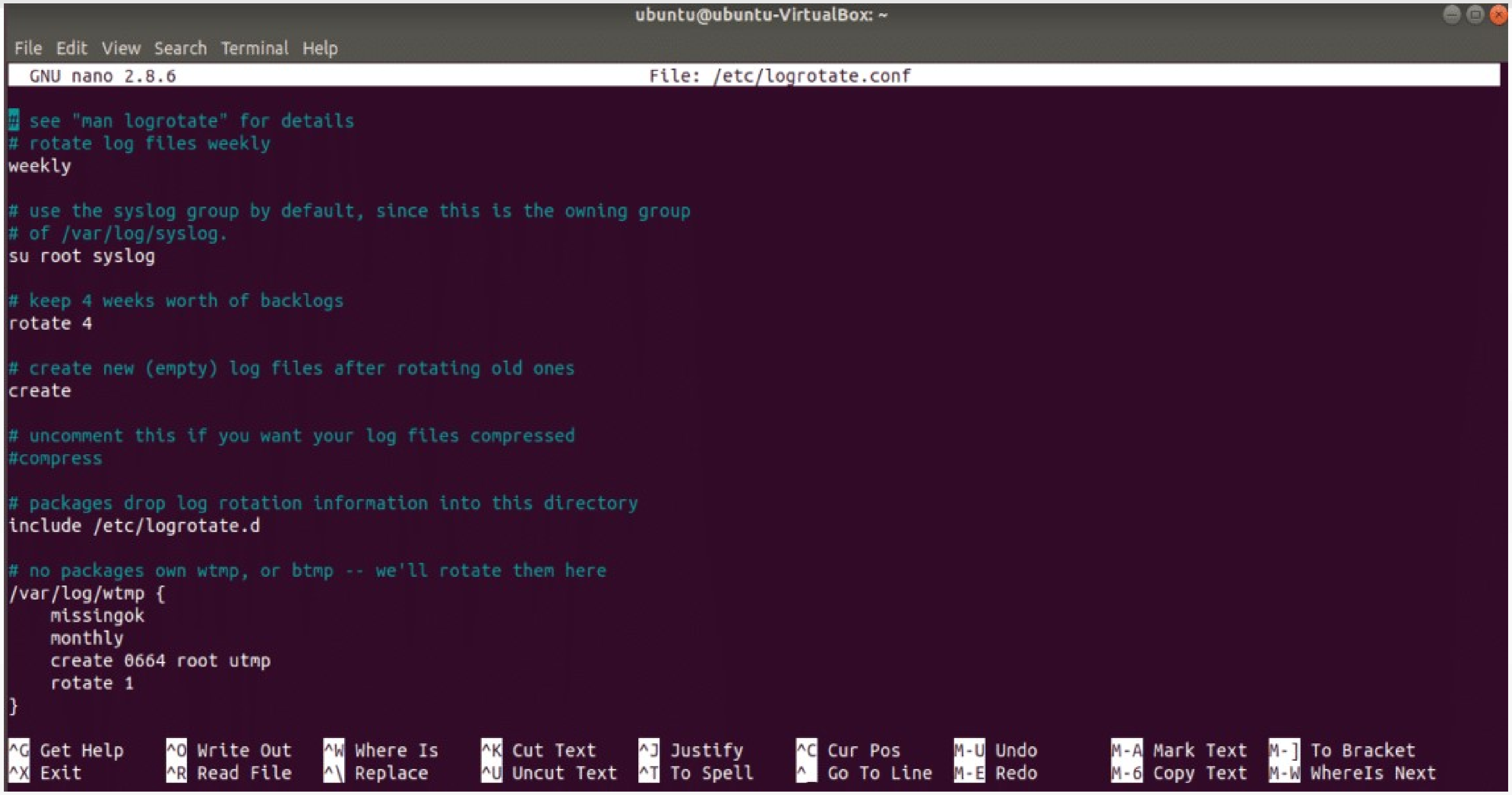
Maintaining logs

- Left unchecked, system logs will grow until you run out of disk space.
- Red Hat Enterprise Linux provides logrotate, a powerful tool for log file maintenance.
- Log files are rotated at predefined intervals.
- logrotate can be configured to compress old log files to conserve filesystem space.

Maintaining logs (cont.)

- Example
  - /var/log/messages is rotated weekly to /var/log/messages.1, with older log files rotated to /var/log/messages.2, etc.
  - Generally, four weeks of log files are kept; older rotated files are deleted.
- When you install a new system service by Red Hat Package Manager (RPM) that generates its own log file, it should "pre-configure" itself automatically for log file rotation by placing a configuration file in the /etc/logrotate.d directory.

A sample of logrotate.conf File


ubuntu@ubuntu-VirtualBox: ~
File Edit View Search Terminal Help

```
  GNU nano 2.8.6                                File: /etc/logrotate.conf

# see "man logrotate" for details
# rotate log files weekly
weekly

# use the syslog group by default, since this is the owning group
# of /var/log/syslog.
su root syslog

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo      M-A Mark Text  M-] To Bracket
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line M-E Redo      M-6 Copy Text  M-W WhereIs Next
```

Maintaining Logs

- Monitoring system logs is an onerous but important task.
  - If you do not properly monitor your logs, you might miss
    - Security problems
    - Hardware problems
    - Software problems

Maintaining Logs

- The logwatch utility, can be installed, parses log files and generates a report that can either be printed or sent to a user via e-mail.
- By default, logwatch runs nightly as a result of the 00-logwatch script located in the /etc/cron.daily directory. The result of this nightly execution is a report e-mailed to the root user.
- The logwatch utility can be configured to detect almost any type of activity with the /etc/logwatch/conf/logwatch.conf file.

* See /usr/share/doc/logwatch-version (where version is the version of logwatch that you are using) for information on writing log filters.

Thanks ☺

SBAHADER@GMAIL.COM