

# CSGE602055 Operating Systems

## CSF2600505 Sistem Operasi

### Week 09: Storage, Firmware, Bootloader, & Systemd

Rahmat M. Samik-Ibrahim (ed.)

University of Indonesia

<https://os.vlsm.org/>

Always check for the latest revision!

REV266 20-Feb-2021

# Operating Systems 211<sup>3)</sup> — PJJ from HOME

## ZOOM: A [Mon (or Wed) 10] — B [Mon (or Wed) 15] — C [Tue (or Thu) 08]

Week	Schedule & Deadline <sup>1)</sup>	Topic	OSC10 <sup>2)</sup>
Week 00	22 Feb - 28 Feb 2021	Overview 1, Virtualization & Scripting	Ch. 1, 2, 18.
Week 01	01 Mar - 07 Mar 2021	Overview 2, Virtualization & Scripting	Ch. 1, 2, 18.
Week 02	15 Mar - 21 Mar 2021	Security, Protection, Privacy, & C-language.	Ch. 16, 17.
Week 03	22 Mar - 28 Mar 2021	File System & FUSE	Ch. 13, 14, 15.
Week 04	29 Mar - 04 Apr 2021	Addressing, Shared Lib, & Pointer	Ch. 9.
Week 05	05 Apr - 11 Apr 2021	Virtual Memory	Ch. 10.
Week 06	26 Apr - 02 May 2021	Concurrency: Processes & Threads	Ch. 3, 4.
Week 07	03 May - 09 May 2021	Synchronization & Deadlock	Ch. 6, 7, 8.
Week 08	24 May - 30 May 2021	Scheduling + W06/W07	Ch. 5.
Week 09	24 May - 06 Jun 2021	Storage, Firmware, Bootloader, & Systemd	Ch. 11.
Week 10	07 Jun - 13 Jun 2021	I/O & Programming	Ch. 12.

<sup>1)</sup> The **DEADLINE** of Week 00 is 28 Feb 2021, whereas the **DEADLINE** of Week 01 is 07 Mar 2021, and so on...

<sup>2)</sup> Silberschatz et. al.: **Operating System Concepts**, 10<sup>th</sup> Edition, 2018.

<sup>3)</sup> This information will be on **EVERY** page two (2) of this course material.

# STARTING POINT — <https://os.vlsm.org/>

- ❑ **Text Book** — Any recent/decent OS book. Eg. (**OSC10**) Silberschatz et. al.: **Operating System Concepts**, 10<sup>th</sup> Edition, 2018. See also <http://codex.cs.yale.edu/avi/os-book/OS10/>.
- ❑ **Resources**
  - ❑ **SCELE OS211** — <https://scele.cs.ui.ac.id/course/view.php?id=3134>.  
The enrollment key is **XXX**.
  - ❑ **Download Slides and Demos from GitHub.com**  
<https://github.com/UI-FASILKOM-OS/SistemOperasi/>:  
os00.pdf (W00), os01.pdf (W01), os02.pdf (W02), os03.pdf (W03),  
os04.pdf (W04), os05.pdf (W05), os06.pdf (W06), os07.pdf (W07),  
os08.pdf (W08), os09.pdf (W09), os10.pdf (W10).
  - ❑ **Problems** — <https://rms46.vlsm.org/2/>:  
195.pdf (W00), 196.pdf (W01), 197.pdf (W02), 198.pdf (W03),  
199.pdf (W04), 200.pdf (W05), 201.pdf (W06), 202.pdf (W07),  
203.pdf (W08), 204.pdf (W09), 205.pdf (W10).
  - ❑ **LFS** — <http://www.linuxfromscratch.org/lfs/view/stable/>
  - ❑ **OSP4DISS** — <https://osp4diss.vlsm.org/>
  - ❑ **DOIT** — <https://doit.vlsm.org/001.html>

# Agenda

- 1 Start
- 2 Schedule
- 3 Agenda
- 4 Week 09
- 5 Storage, Firmware, Bootloader, & Systemd
- 6 Storage Management
- 7 RAID
- 8 Legacy BIOS
- 9 UEFI
- 10 Operating System (Boot) Loader
- 11 GRUB Map
- 12 init (SYSV legacy)
- 13 UpStart - Ubuntu
- 14 The All New "systemd"
- 15 systemctl
- 16 Linux From Scratch (part 02)
- 17 Week 09: Check List

# Week 09 Storage, Firmware, Bootloader, & Systemd: Topics<sup>1</sup>

- Storage
- Storage Arrays
- BIOS
- Loader
- Systemd

---

<sup>1</sup>Source: ACM IEEE CS Curricula 2013

# Week 09 Storage, Firmware, Bootloader, & Systemd: Learning Outcomes<sup>1</sup>

- Storage [Usage]
- Storage Arrays [Usage]
- BIOS [Usage]
- Loader [Usage]
- Systemd [Usage]

---

<sup>1</sup>Source: ACM IEEE CS Curricula 2013

# Storage, Firmware, Bootloader, & Systemd

- Reference: (OSC10-ch11)
- Storage Capacity (2019)<sup>1</sup>
  - Legacy 3.5" Floppy Disk (1.4MB) – obsolete?
    - SuperDisk (up to 240 MB) — never took off.
  - 4.7" Compact Disc (700MB) – obsolete?
    - 4.7" Digital Versatile Disc (up to 9GB) – ?
    - 4.7" Blu Ray (up to 128 GB) ⇒ DVD++.
  - Tape Cartridge (up to 15TB)
    - Robotic System (up to 250 PB per unit)
    - NASA, Google, Microsoft are still using this!
    - Cheap but slow.
  - Hard Disk Drives (up to 16 TB).
    - From Perpendicular Magnetic Recording to Shingled Magnetic Recording technology (+25% – writing problems).
    - Mechanical Disk Arm Scheduling (Until When?).
  - Solid-State Disks (up to 16 TB).
    - SSD Price > HDD Price.
    - Write Speed >> Read Speed.
    - (What is a) Flash Disk?

---

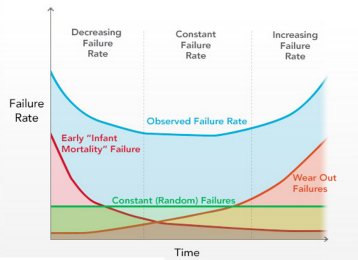
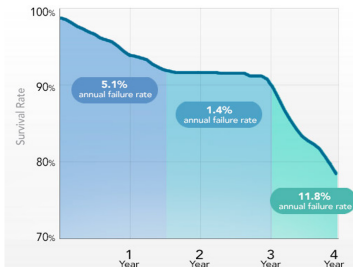
<sup>1</sup>Subject to change

# Storage Failure Rates

- MTDDL: Mean Time To Data Loss
- MTTF: Mean Time To Failure
- BackBlaze (Cloud Backup Services)

Drives Have 3 Distinct Failure Rates      General Predicted Failure Rates

Hard Drive Survival Rates - Chart 1



<https://www.extremetech.com/computing/170748-how-long-do-hard-drives-actually-live-for>



Figure: BackBlaze — Failure Rates of 25000 DISKS



# Storage Management

- Attached-Storage.
  - Host-Attached Storage: via I/O.
  - Network-Attached Storage (NAS): via distributed FileSystem.
  - Storage Area Network (SAN): dedicated Network.
- Formatting
  - Low Level (Physical)
  - High Level (FileSystem)
- Boot Block
- Disk Partition
  - "MBR"-scheme
    - upto 4 primary partition
    - upto 2 TB disk
  - "GPT"-scheme
    - "unlimited" partition
    - "unlimited" disk
    - redundancy
- Swap Space Management: On Partition or FileSystem?

# RAID: Redundant Array of In\* Disks

- RAID 0, 1, 5, 6, 10, 100
- Note (<http://www.commodore.ca/windows/raid5/raid5.htm>):
  - RAID was created to enhance data performance, reliability and availability.
  - Striping, parity checking and mirroring are three primary functions of RAID systems.
  - RAID performs its functions transparent to the operating system.
  - Systems are typically defined by ranks consisting of five disks each connected to one or two Disk Array Controllers.
  - Different RAID levels provide varying degrees of speed and data protection.
- Problems with RAID
- Stable-Storage Implementation

# BIOS, Boot, & Systemd

- Firmware
  - BIOS: Basic Input Output System.
  - UEFI: Unified Extensible Firmware Interface.
  - ACPI: Advanced Configuration and Power Interface.
- Operating System (Boot) Loader
  - BOOTMGT: Windows Bootmanager / Bootloader.
  - LILO: Linux Loader.
  - GRUB: GRand Unified Bootloader.
- Operating System Initialization
  - Init (legacy)
  - UpStart
  - Systemd

- Check Settings.
- Initialize CPU & RAM.
- POST: Power-On Self-Test.
- Initialize ports, LANS, etc.
- Load a Boot Loader.
- Handover to the Boot Loader.
- Provides "Native" (obsolete) Drivers only (not loadable).
- Provides "INT" services .
- Limitation.
  - Technology of 1970s.
  - 16 bits software.
  - 20 bits address space (1 MB).
  - 31 bits disk space (2 TB).

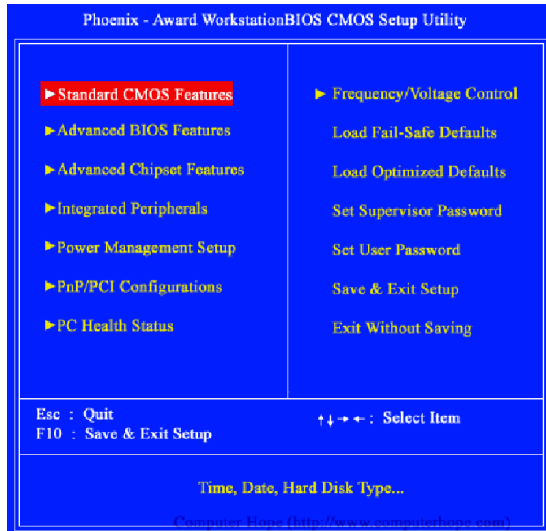


Figure: BIOS

- A Firmware Specification, not an Implementation!
- No (INT) service after boot.
- HII: Human Interface Infrastructure.
- Protected Mode.
- Flexible.
  - Technology of 2000s.
  - written in C.
  - (third party) loadable drivers and tools.
  - Emulate Legacy BIOS transition (MBR block, INT service).
  - UEFI Shell: environment shell for diagnostic (no need for DOS).
- Problems
  - Who controls the Hardware?
  - Is "Secure Boot" a good thing?
  - How about a **NASTY/LOCKING/TROJAN** UEFI implementation?
  - Different **DRIVERS**.

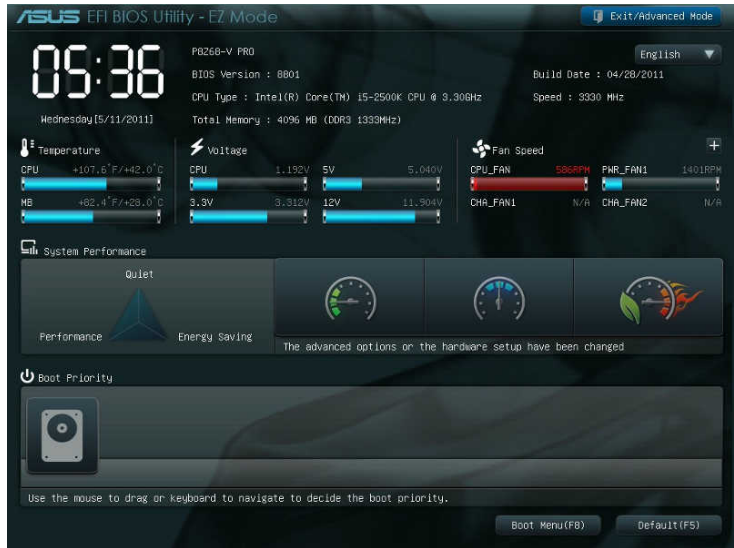


Figure: UEFI

## Platform Initialization (PI) Boot Phases

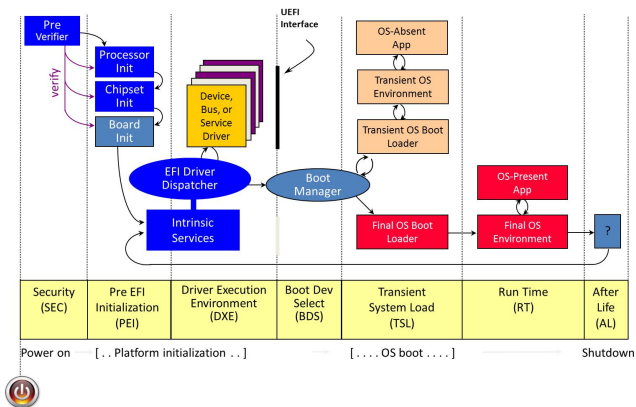


Figure: UEFI Boot Process<sup>1</sup>.

<sup>1</sup>Source Jarslstrom - 2014 - [www.tianocore.org](http://www.tianocore.org)



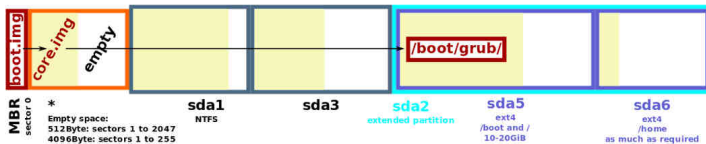
# Operating System (Boot) Loader

- General
  - How/Where to start the operating system?
  - What to do?
  - How many ways to boot?
  - How many types of OS?
- Disk Partition
  - MBR: Master Boot Record (1983).
  - GPT: GUID (Globally Unique Identifiers) Partition Table (2010s).
- GRUB: GRand Unified Boot system
  - Stage 1: a small boot.img inside the MBR.
  - Stage 1.5 (core.img): FileSystem drivers after MBR.
  - Stage 2: Kernel Selection: Windows, Linux, BSD, etc.
- GRUB2
  - More flexible than GRUB legacy.
  - More automated than GRUB legacy.
  - Accept MBR and GPT.
  - Stage 1.5 (core.img): generated from diskboot.img.
  - No 1024 cylinder restriction.

## GNU GRUB 2

Locations of *boot.img*, *core.img* and the */boot/grub* directory

Example 1: an MBR-partitioned harddisc with sector size of 512 or 4096Bytes



Example 2: a GPT-partitioned harddisc with sector size of 512 or 4096Bytes

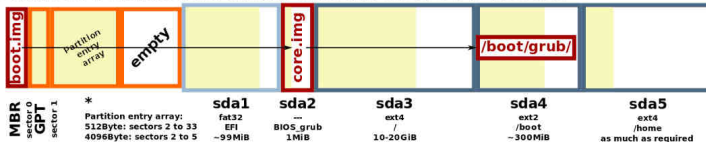


Figure: GRUB<sup>1</sup>.

<sup>1</sup>Source Shmuel Csaba Otto Traian 2013

# init (SYSV legacy)

- File: `/etc/inittab`.
- Folders: `/etc/rcX.d` — `X` = runlevel.
  - Seven (7) different runlevels:
    - 0 (shutdown).
    - 1 (single-user/admin).
    - 2 (multi-user non net).
    - 3 (standard).
    - 4 (N/A).
    - 5 (3+GUI).
    - 6 (reboot).
  - `SXX-YYY`: Start
  - `KXX-YYY`: Kill.
- One script at a time in order.
- dependency is set manually.

- Developer: Ubuntu.
- Folder: `/etc/init/`.
- Control: `initctl`.
  - `initctl list` – listing all processes managed by upstart.
- better support for hotplug devices.
- cleaner service management.
- faster service management.
- asynchronous.

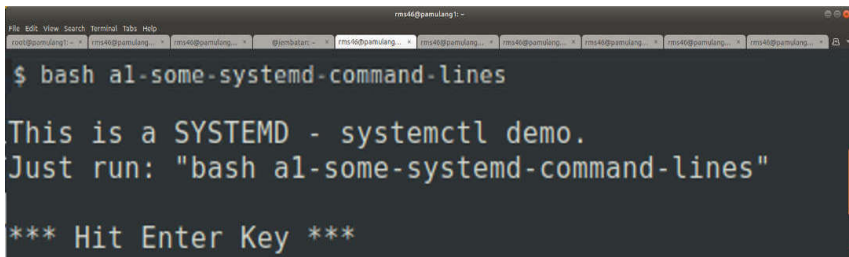
# The All New "systemd"

- Replaces (SYSV) init and UpStart.
  - better concurrency handling: Faster!
  - better dependencies handling: No more "S(tarts)" and "K(ills)".
  - better crash handling: automatic restart option.
  - better security: group protection from anyone including superusers.
  - simpler config files: reliable and clean scripts.
  - hotplug: dynamic start/stop.
  - supports legacy systems (init).
  - overhead reducing.
  - unified management way for all distros.
  - bloated: doing more with more resources.
  - linux specific: NOT portable.

# systemctl 01

```
for II in \
'systemctl list-unit-files | head -8; echo "(...)";
  systemctl list-unit-files| tail -8' \
'systemd-analyze blame | wc -l; echo "===";
  systemd-analyze blame | head -15' \
'systemctl --full | wc -l; echo "===";
  systemctl --full | head -10' \
'systemctl list-units | wc -l; echo "===";
  systemctl list-units | head -10' \
'systemctl list-units |grep .service|wc -l;echo "===";
  systemctl list-units|grep .service|head -10' \
'systemctl list-units | grep ssh.service' \
'systemctl status ssh.service' \
'systemctl is-enabled ssh' \
'journalctl' \
'journalctl -b' \
do
...
```

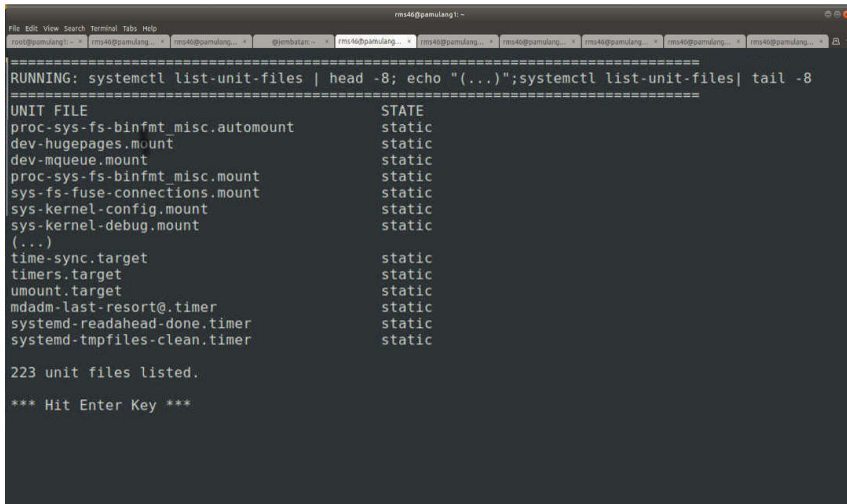
# systemctl 02

A terminal window titled 'rms46@pamulang1: -' with multiple tabs. The active tab shows the command '\$ bash a1-some-systemd-command-lines' and its output: 'This is a SYSTEMD - systemctl demo. Just run: "bash a1-some-systemd-command-lines" \*\*\* Hit Enter Key \*\*\*'.

```
rms46@pamulang1: -  
root@pamulang1:~$ bash a1-some-systemd-command-lines  
This is a SYSTEMD - systemctl demo.  
Just run: "bash a1-some-systemd-command-lines"  
*** Hit Enter Key ***
```

Figure: bash a1-some-systemd-command-lines

# systemctl 03

A terminal window titled 'rms46@pamulang1: ~' with multiple tabs. The active tab shows the command 'systemctl list-unit-files | head -8; echo "...";systemctl list-unit-files| tail -8' being executed. The output lists unit files and their states. The units shown are: proc-sys-fs-binfmt\_misc.automount (static), dev-hugepages.mount (static), dev-mqueue.mount (static), proc-sys-fs-binfmt\_misc.mount (static), sys-fs-fuse-connections.mount (static), sys-kernel-config.mount (static), sys-kernel-debug.mount (static), (...), time-sync.target (static), timers.target (static), umount.target (static), mdadm-last-resort@.timer (static), systemd-readahead-done.timer (static), and systemd-tmpfiles-clean.timer (static). The terminal also indicates that 223 unit files are listed in total and prompts the user to hit the Enter key.

```
=====
RUNNING: systemctl list-unit-files | head -8; echo "...";systemctl list-unit-files| tail -8
=====
UNIT FILE                                     STATE
proc-sys-fs-binfmt_misc.automount           static
dev-hugepages.mount                         static
dev-mqueue.mount                           static
proc-sys-fs-binfmt_misc.mount               static
sys-fs-fuse-connections.mount               static
sys-kernel-config.mount                    static
sys-kernel-debug.mount                     static
(...)
time-sync.target                           static
timers.target                             static
umount.target                             static
mdadm-last-resort@.timer                   static
systemd-readahead-done.timer               static
systemd-tmpfiles-clean.timer               static

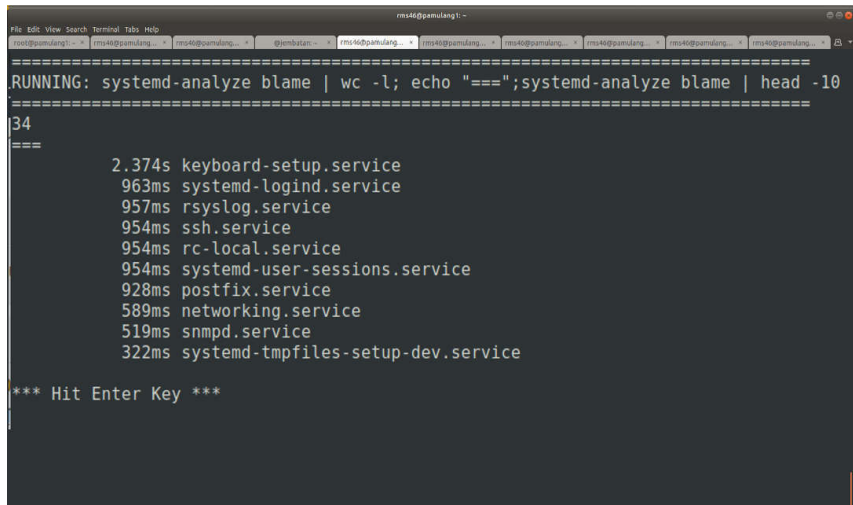
223 unit files listed.

*** Hit Enter Key ***
```

Figure: systemctl list-unit-files



# systemctl 04

A terminal window titled 'rms46@pamulang1: ~' showing the output of the command 'systemd-analyze blame'. The output lists various system services and their boot times. The command is preceded by 'RUNNING: ' and followed by a pipe to 'wc -l; echo "===";systemd-analyze blame | head -10'. The output shows a list of services with their respective boot times in seconds and milliseconds. The services listed are: keyboard-setup.service (2.374s), systemd-logind.service (963ms), rsyslog.service (957ms), ssh.service (954ms), rc-local.service (954ms), systemd-user-sessions.service (954ms), postfix.service (928ms), networking.service (589ms), snmpd.service (519ms), and systemd-tmpfiles-setup-dev.service (322ms). The output is preceded by '34' and '===', and followed by '\*\*\* Hit Enter Key \*\*\*'.

```
File Edit View Search Terminal Tabs Help
rms46@pamulang1: ~
RUNNING: systemd-analyze blame | wc -l; echo "===";systemd-analyze blame | head -10
34
===
2.374s keyboard-setup.service
963ms systemd-logind.service
957ms rsyslog.service
954ms ssh.service
954ms rc-local.service
954ms systemd-user-sessions.service
928ms postfix.service
589ms networking.service
519ms snmpd.service
322ms systemd-tmpfiles-setup-dev.service
*** Hit Enter Key ***
```

Figure: systemd-analyze blame

# systemctl 05

```
rms40@pamulang1: ~
File Edit View Search Terminal Tabs Help
rms40@pamulang1: ~
RUNNING: systemctl --full | wc -l; echo "===";systemctl --full | head -6
=====
97
UNIT
proc-sys-fs-binfmt-misc.automount                                loaded active waiting Arbitrary Executable File Formats File System Automount Point
sys-devices-pci0000:00-0000:00:05.0-host0-target0:0:0:0:0-block-sda-sda1.device loaded active plugged QEMU_HARDDISK 1
sys-devices-pci0000:00-0000:00:05.0-host0-target0:0:0:0:0-block-sda-sda2.device loaded active plugged QEMU_HARDDISK 2
sys-devices-pci0000:00-0000:00:05.0-host0-target0:0:0:0:0-block-sdb.device loaded active plugged QEMU_HARDDISK
sys-devices-pci0000:00-0000:00:05.0-host0-target0:0:1:0:0:1-block-sdb-sdb1.device loaded active plugged QEMU_HARDDISK 1
*** Hit Enter Key ***

RUNNING: systemctl list-units | wc -l; echo "===";systemctl list-units | head -6
=====
97
UNIT
proc-sys-fs-binfmt-misc.automount                                loaded active waiting Arbitrary Executable File Formats File System Automount Point
sys-devices-pci0000:00-0000:00:05.0-host0-target0:0:0:0:0-block-sda-sda1.device loaded active plugged QEMU_HARDDISK 1
sys-devices-pci0000:00-0000:00:05.0-host0-target0:0:0:0:0-block-sda-sda2.device loaded active plugged QEMU_HARDDISK 2
sys-devices-pci0000:00-0000:00:05.0-host0-target0:0:0:0:0-block-sdb.device loaded active plugged QEMU_HARDDISK
sys-devices-pci0000:00-0000:00:05.0-host0-target0:0:1:0:0:1-block-sdb-sdb1.device loaded active plugged QEMU_HARDDISK 1
*** Hit Enter Key ***

RUNNING: systemctl list-units | grep .service | wc -l; echo "===";systemctl list-units | grep .service | head -6
=====
12
UNIT
ssct.service                                                    loaded active exited LSB: process and login accounting
icpidd.service                                                  loaded active running ACPI event daemon
console-setup.service                                           loaded active exited Set console font and keymap
cron.service                                                    loaded active running Regular background program processing daemon
dbus.service                                                    loaded active running D-Bus System Message Bus
jetty@tty1.service                                              loaded active running Getty on tty1
*** Hit Enter Key ***
```

Figure: systemctl --full; systemctl list-units

# systemctl 06

```
rms46@pamulang1: ~
File Edit View Search Terminal Tabs Help
rms46@pamulang1 ~
=====
RUNNING: systemctl list-units | grep ssh.service
=====
ssh.service                                loaded active running    OpenBSD Secure Shell server
*** Hit Enter Key ***
=====
RUNNING: systemctl status ssh.service
=====
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
   Active: active (running) since Sun 2020-04-26 03:00:24 WIB; 3h 33min ago
   Process: 653 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 686 (sshd)
   CGroup: /system.slice/ssh.service
           └─ 686 /usr/sbin/sshd -D
              └─ 3247 sshd: demo [priv]
                 └─ 3253 sshd: demo@pts/0
                    └─ 3254 -bash
                       └─ 3391 bash a1-some-systemd-command-lines
                          └─ 3550 systemctl status ssh.service
*** Hit Enter Key ***
```

Figure: systemctl status ssh.service

A terminal window titled 'rms46@pamulang1: ~' with multiple tabs. The active tab shows the command 'systemctl status ssh' and its output. The output is displayed in a monospaced font with a dark background and light-colored text. The text is as follows:

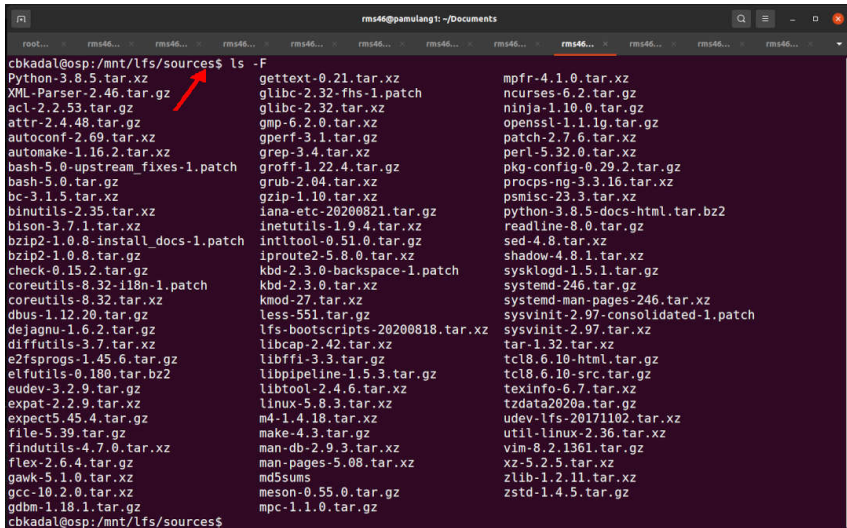
```
=====  
RUNNING: systemctl is-enabled ssh  
=====  
enabled  
  
*** Hit Enter Key ***
```

Figure: systemctl is-enabled ssh

# Linux From Scratch (part 02)

- Week 09 (part 02).
  - Follow the "Linux From Scratch" book, chapter 01-06 (<http://www.linuxfromscratch.org/lfs/view/stable/>).
  - Screen capture with your **OWN** account (e.g. "cbkadal"):
    - ("W09-LFS-1.jpg"): LFS source-codes /mnt/lfs/sources/.
    - ("W09-LFS-2.jpg"): Disk size /dev/sda1/ and /dev/sdb1/.
  - Save (export) the result into "LFSC.ova".
  - Sign it as: "LFSC.ova.asc" (armor, detach).
  - Deliveries: tarball "TXT/myW09.tar.bz2.txt" (signed, asymmetric-key, armor) of folder W09/ (**4 files**).
    - ❶ Blank File: "YourAccount-YourStudentID.txt" (eg. "cbkadal-2006123456.txt").
    - ❷ File: "W09-LFS-1.jpg" (page 30).
    - ❸ File: "W09-LFS-2.jpg" (page 31).
    - ❹ File: "LFSC.ova.asc" (signature, armor, detach).
- Keep but **DO NOT SUBMIT** file LFSC.ova!.
- See also <https://lfs.vlsm.org/>.

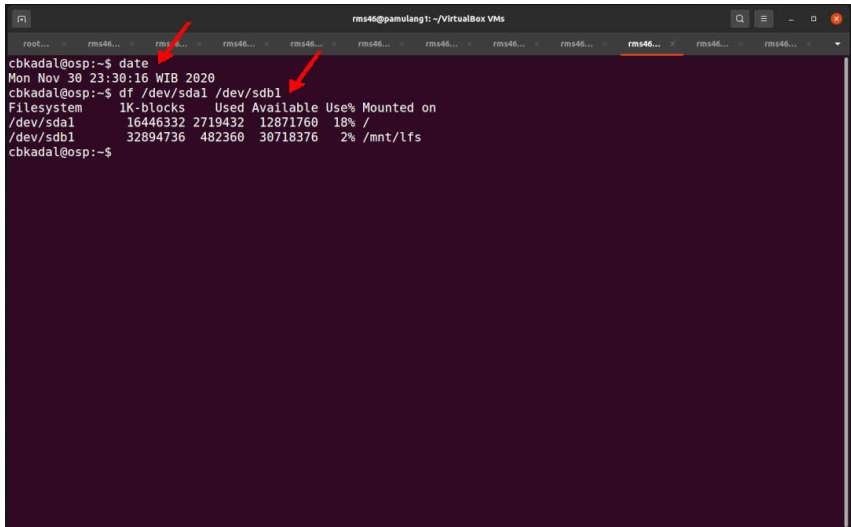
# W09-LFS-1.jpg



```
cbkadal@osp:/mnt/lfs/sources$ ls -F
Python-3.8.5.tar.xz      gettext-0.21.tar.xz      mpfr-4.1.0.tar.xz
XML-Parser-2.46.tar.gz  glibc-2.32-fhs-1.patch  ncurses-6.2.tar.gz
acl-2.2.53.tar.gz       glibc-2.32.tar.xz       ninja-1.10.0.tar.gz
attr-2.4.48.tar.gz      gmp-6.2.0.tar.xz        openssl-1.1.1g.tar.gz
autoconf-2.69.tar.xz    gperf-3.1.tar.gz        patch-2.7.6.tar.xz
automake-1.16.2.tar.xz  grep-3.4.tar.xz         perl-5.32.0.tar.xz
bash-5.0-upstream_fixes-1.patch  groff-1.22.4.tar.gz    pkg-config-0.29.2.tar.gz
bash-5.0.tar.gz         grub-2.04.tar.xz        procs-ng-3.3.16.tar.xz
bc-3.1.5.tar.xz         gzip-1.10.tar.xz        psmisc-23.3.tar.xz
binutils-2.35.tar.xz    iana-etc-20200821.tar.gz  python-3.8.5-docs-html.tar.bz2
bison-3.7.1.tar.xz      inetutils-1.9.4.tar.xz  readline-8.0.tar.gz
bzip2-1.0.8-install_docs-1.patch  intltool-0.51.0.tar.gz  sed-4.8.tar.xz
bzip2-1.0.8.tar.gz      iproute2-5.8.0.tar.xz   shadow-4.8.1.tar.xz
check-0.15.2.tar.gz     kbd-2.3.0-backspace-1.patch  sysklogd-1.5.1.tar.gz
coreutils-8.32-ii8n-1.patch  kbd-2.3.0.tar.xz        systemd-246.tar.gz
coreutils-8.32.tar.xz    kmod-27.tar.xz          systemd-man-pages-246.tar.xz
dbus-1.12.20.tar.gz     less-551.tar.gz         sysvinit-2.97-consolidated-1.patch
dejagnu-1.6.2.tar.gz     lfs-bootscripts-20200818.tar.xz  sysvinit-2.97.tar.xz
diffutils-3.7.tar.xz    libcap-2.42.tar.xz      tar-1.32.tar.xz
e2fsprogs-1.45.6.tar.gz  libffi-3.3.tar.gz       tcl8.6.10-html.tar.gz
elfutils-0.180.tar.bz2  libpipeline-1.5.3.tar.gz  tcl8.6.10-src.tar.gz
eudev-3.2.9.tar.gz      libtool-2.4.6.tar.xz    texinfo-6.7.tar.xz
expat-2.2.9.tar.xz      linux-5.8.3.tar.xz      tzdata2020a.tar.gz
expect5.45.4.tar.gz     m4-1.4.18.tar.xz        udev-lfs-20171102.tar.xz
file-5.39.tar.gz        make-4.3.tar.gz         util-linux-2.36.tar.xz
findutils-4.7.0.tar.xz  man-db-2.9.3.tar.xz     vim-8.2.1361.tar.gz
flex-2.6.4.tar.gz       man-pages-5.08.tar.xz   xz-5.2.5.tar.xz
gawk-5.1.0.tar.xz       md5sums                 zlib-1.2.11.tar.xz
gcc-10.2.0.tar.xz       meson-0.55.0.tar.gz     zstd-1.4.5.tar.gz
gdbm-1.18.1.tar.gz      mpc-1.1.0.tar.gz
cbkadal@osp:/mnt/lfs/sources$
```

Figure: W09-LFS-1.jpg

# W09-LFS-2.jpg



A terminal window titled "rms46@pamulang1: ~/VirtualBox VMs" showing the output of the 'date' and 'df' commands. The 'date' command shows the current date and time as "Mon Nov 30 23:30:16 WIB 2020". The 'df' command shows the disk usage for /dev/sda1 and /dev/sdb1. The output for /dev/sda1 shows 16446332 1K-blocks used, 12871760 available, and 18% usage. The output for /dev/sdb1 shows 32894736 1K-blocks used, 30718376 available, and 2% usage. Two red arrows point to the 'date' and 'df' commands in the terminal output.

```
rms46@pamulang1: ~/VirtualBox VMs
root... rms46... rms46... rms46... rms46... rms46... rms46... rms46... rms46... rms46... rms46... rms46...
cbkadal@osp:~$ date
Mon Nov 30 23:30:16 WIB 2020
cbkadal@osp:~$ df /dev/sda1 /dev/sdb1
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1       16446332 2719432  12871760  18% /
/dev/sdb1       32894736 482360  30718376   2% /mnt/lfs
cbkadal@osp:~$
```

Figure: W09-LFS-2.jpg

# Week 09: Check List (Deadline: Monday, 07-Dec-2020).

## □ Week 09: Linux From Scratch (part 2)(os09.pdf).

- ① Week 09 Token: **ROMULAN**
- ② Read: (OSC10 chapter 11)
- ③ Update your Virtual Guest.
- ④ Visit <https://os.vlsm.org/GitHubPages/>. Review **Last Week TOP 10 List** and pick at least 3 out of your 10 closest neighbors. See <https://cbkadal.github.io/os211/TXT/myrank.txt>.
- ⑤ Create your **TOP 10 List** of Week 09 (e.g. <https://cbkadal.github.io/os211/W09/>). **Do not use lecture material. Please be more creative!**
- ⑥ Run "chktoken ROMULAN" and write the result into myW09token.txt.
- ⑦ Update your log (e.g. <https://cbkadal.github.io/os211/TXT/mylog.txt>).
- ⑧ Update bash script (e.g. <https://cbkadal.github.io/os211/TXT/myscript.sh>).
- ⑨ Make SHA256SUM and sign it (detached, armor) as SHA256SUM.asc.
- ⑩ Place the assignment result into W09/ folder and "tar" it into myW09.tar.bz2.
- ⑪ GnuPG encrypt (armor) to myW09.tar.bz2.txt.



# The End

- ☐ This is the end of the presentation.
- ☒ This is the end of the presentation.
  - This is the end of the presentation.