CSGE602055 Operating Systems
CSF2600505 Sistem Operasi
Week 01: Overview 2, Virtualization & Scripting

C. BinKadal

Sendirian Berhad

https://os.vlsm.org/Slides/os01.pdf
Always check for the latest revision!

REV391 19-Aug-2022

OS222[3]): Operating Systems Schedule 2022 - 2

Week	Topic[1])	OSC10[2])
Week 00	Overview (1), Assignment of Week 00	Ch. 1, 2
Week 01	Overview (2), Virtualization & Scripting	Ch. 1, 2, 18.
Week 02	Security, Protection, Privacy, & C-language.	Ch. 16, 17.
Week 03	File System & FUSE	Ch. 13, 14, 15.
Week 04	Addressing, Shared Lib, & Pointer	Ch. 9.
Week 05	Virtual Memory	Ch. 10.
Week 06	Concurrency: Processes & Threads	Ch. 3, 4.
Week 07	Synchronization & Deadlock	Ch. 6, 7, 8.
Week 08	Scheduling + W06/W07	Ch. 5.
Week 09	Storage, Firmware, Bootloader, & Systemd	Ch. 11.
Week 10	I/O & Programming	Ch. 12.

[1]) For schedule, see https://os.vlsm.org/#idx02

[2]) Silberschatz et. al.: **Operating System Concepts**, 10^{th} Edition, 2018.

[3]) 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^{th} Edition, 2018. (See https://www.os-book.com/OS10/).
- **Resources** (https://os.vlsm.org/#idx03)
  - **SCELE OS222** — https://scele.cs.ui.ac.id/course/view.php?id=3398. The enrollment key is **XXX**.
  - **Download Slides and Demos from GitHub.com** — (https://github.com/os2xx/os/) 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** 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/
  - **This is How Me Do It!** — https://doit.vlsm.org/001.html
    - PS: "Me" rhymes better than "I", duh!

Agenda

Week 01 Overview II: Topics[1]

- Intelectual Property Rights (IPR)
- Software Licenses and Free Software
- Operating System Services and Interfaces
- System Calls and System Programming
- Types of virtualization (including Hardware/Software, OS, Server, Service, Network)
- Hypervisors
- Portable and cost of virtualization; emulation vs. isolation
- Cloud services: IAAS, PAAS and Platform APIs, SAAS
- Introduction to Scripting and REGEX.

[1]Source: ACM IEEE CS Curricula 2013

Week 01 Overview II: Learning Outcomes[1]

- Explain the concept of virtual memory and how it is realized in hardware and software. [Familiarity]
- Discuss hypervisors and the need for them in conjunction with different types of hypervisors. [Usage]
- Differentiate emulation and isolation. [Familiarity]
- Evaluate virtualization trade-offs. [Assessment]
- Discuss the importance of elasticity and resource management in cloud computing. [Familiarity]
- Explain the advantages and disadvantages of using the virtualized infrastructure. [Familiarity]

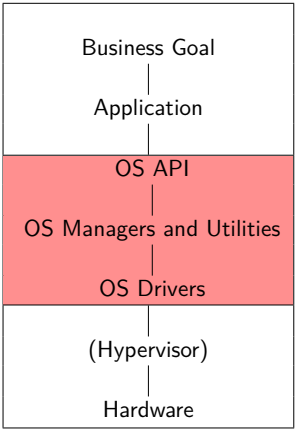
[1]Source: ACM IEEE CS Curricula 2013

OSC10 (Silberschatz) Chapter 18: Virtual Machines

- OSC10 Chapter 18
  - Overview
  - History
  - Benefits and Features
  - Building Blocks
  - Types of Virtual Machines and Their Implementations
  - Virtualization and Operating-System Components
  - Examples

What defines an Operating System? (The Three Layers Model)

URL: https://rahmatm.samik-ibrahim.vlsm.org/2021/07/what-defines-operating-system.html



- The Three Layers Model
  - An Operating System is between your Application and your Hardware (or Hypervisor).
    - OS API: Application Programming Interface
    - OS Resources Managers and Utilities: Process, Scheduler, Dispatcher, (Virtual) Memory, Disk, I/O, Network, Security, Protection, etc.
    - OS Device Drivers: controls devices
  - Remember that your future "**Business Goal**" may not directly relate to an Operating System at all!

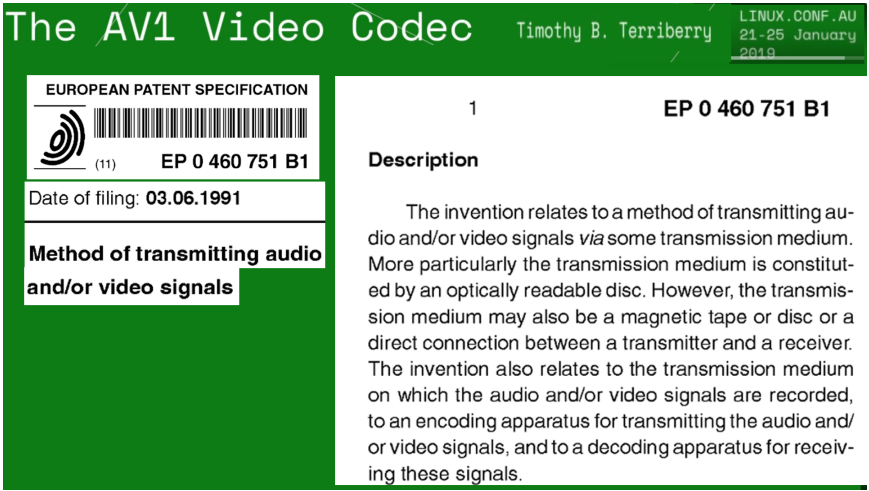
Week 01: Review 2

- Intellectual Property Rights (IPR)
- Richard Stallman: Introduction to Free Software
  - YouTube: https://youtu.be/Ag1AKIl_2GM (article).
  - See also https://rms46.vlsm.org/1/70.pdf
- Operating System Services
- User Operating System Interface
- System Calls
- Types of System Calls
- System Programs
- Operating System Design and Implementation
- Operating System Structure
- Introduction to GNU/Linux.
- More Operating Systems.

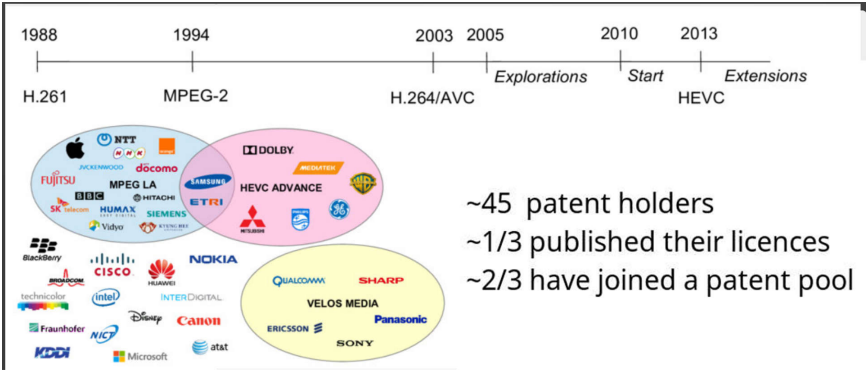
Intelectual Property Rights (IPR)

- Trade Secret (Rahasia Dagang) — UU no. 30/2000.
- Industrial Design (Desain Industri) — UU no. 31/2000.
- Integrated Circuit Layout Design (Desain Tata Letak Sirkuit Terpadu) — UU no. 32/2000.
- Patent (Paten) — UU no. 14/2001.
- Copyright (Hak Cipta) — UU no. 19/2002.
- The problem of Intellectual Property Rights (IPR).
- Software IPR.
- Software Licenses: GNU GPL, EULA, Public Domain, Apache, Microsoft Public License.

Is this a Software Patent or Not?



The Codec Mess



~45 patent holders
~1/3 published their licences
~2/3 have joined a patent pool

Courtesy of
Jonatan Samuelsson
Divideon
Co-founder and CEO

Alliance for Open Media

Source (per 21-Sep-2020): https://aomedia.org/membership/members/

Free Software

- Free Software Definition (FSF)
  0. The freedom to run the program as you wish, for any purpose (freedom 0).
  1. The freedom to study how the program works and change it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
  2. The freedom to redistribute copies so you can help your neighbor (freedom 2).
  3. The freedom to distribute copies of your modified versions to others (freedom 3). By doing this, you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.
- Free Software vs. Open Source Software.
- Copyleft Software.

Software Licenses

- 3-clause BSD license and 2-clause BSD license (BSD-X-Clause)
- Apache License 2.0 (Apache-2.0)
- Artistic License 2.0 (ArtisticLicense2)
- Common Development and Distribution License (CDDL-1.0)
- Eclipse Public License (EPL-1.0)
- Educational Community License 2.0 (ECL2.0)
- Expat License (Expat) aka. MIT license (MIT)
- GNU Affero General Public License v3 (AGPL-3.0)
- GNU All-Permissive License (GNUAllPermissive)
- GNU General Public License (GPL)
- GNU Lesser General Public License (LGPL)
- Microsoft Public License (MS-PL)
- Mozilla Public License 2.0 (MPL-2.0)
- "Public Domain" (PublicDomain)
- X11 License (X11License)

Virtualization & Cloud Computing

- Virtual Machine
  - Host & Guest
  - Hypervisor (Virtual Machine Manager)
    - Type 0, 1, 2 Hypervisor
    - ParaVirtualization
    - Programming-environment Virtualization
    - Emulators
  - Application Containment (OS-Level)
    - Containers: LXC, Solaris Containers, Docker.
    - Zones: Solaris Containers
    - Virtual Private Servers: OpenVZ
    - Virtual Kernels: DragonFly BSD
    - Jails: FreeBSD Jail/ Chroot Jail
  - Kubernetes (K8s): A (open source) system for managing CONTAINERIZED applications.
- Cloud Computing
  - SAAS: Software As A Service.
  - PAAS: Platform As A Service.
  - IAAS: Infrastructure As A Service.

Potpourri

- Mobile/Distributed/Client-Server/Peer-to-Peer Computing.
- Real-Time Computing: Hard Real-Time vs. Soft Real-Time.
- Operating System Comparison: Android, *BSD, GNU/Linux, iOS, Mac OS, Windows.
- Operating System Services: UI (GUI, CLI); Program Executing; I/O Operations; File Systems Manipulation; Communication; Error Detection; Resource Allocation; Accounting; Protection & Security.
- System Calls: Process Control; File Management; Device Management; Information Maintenance; Communications; Protection.
- Application Programming Interface (API)
- Standard C Library.
- System Programs.
- Microkernel System Structure.
- Loadable Kernel Modules.
- Virtualization and Cloud System.

Some Essential Command Lines (1)

man	manual. E.g., `"man man"`
passwd	changes passwords.
ls	list directory contents. E.g., `"ls -al"`
cd	change the working directory. E.g., `"cd /tmp"`
cp	copy file(s). E.g., `"cp SOURCE DEST"`
rm	remove file(s). E.g., `"rm AFILE"`
mv	move files(s). E.g., `"mv FROMFILE TOFILE"`
mkdir	make directories(s). E.g., `"mkdir ADIRECTORY"`
rmdir	remove directories(s). E.g., `"rmdir ADIRECTORY"`
cat	read file(s) E.g., `"cat AFILE"`
more	read file(s) per screen E.g., `"more AFILE"`
ln	make a link of a file. E.g., `"ln -s file sfile"`
grep	search string "aword" inside file. E.g., `"grep aword file"`
sort	sort lines of text files. E.g., `"sort file1.txt"`
top	display systems task. E.g., `"top"`

Some Essential Command Lines (2)

find	E.g., "`find / -name minix3.iso -print`". Find from "/".
chmod	E.g. "`chmod 755 file`". Change file with access mode 755.
chown	E.g. "`chown user file`". Change owner file to user.
chgrp	E.g. "`chgrp other file`". Change group file to other.
tar	tape archive file. E.g.
	"`tar cf /tmp/tfile.tar dir/`". Archive "dir/" into tfile.tar.
	"`tar tf /tmp/tfile.tar`". List tfile.tar.
	"`tar xf /tmp/tfile.tar`". Extract tfile.tar.
date	print or set the system date and time. E.g. "`date +%Y`"
tee	read from standard input and write to standard output and files.
	E.g. "`ls -al | tee listing.txt`"
diff	compare files line by line. E.g. "`diff file1.txt file2.txt`"
wc	print newline, word, and byte counts for each file. E.g. "`wc file.txt`"

The "vi" editor

- VI Basics

	Basics		More Commands
i	insert mode	d^	delete from ^ (beginning) to the cursor
a	append mode	d$	delete from the cursor to $ (end)
<ESC>	escape mode	dd	delete the whole line
q!	quit	5dd	delete 5 lines
wq!	write and quit	yy	yank (copy) the line
ZZ	write and quit	p	put (paste) the line
h j k l	move [left, down, up, right]	J	joint current and next line
r	replace a character	:r file.txt	read (insert) file.txt
d	delete a character	:w! file.txt	write into file.txt
u	undo	:1,8 w! file.txt	write line 1 to 8 into file.txt

  - Basic vi Commands https://www.cs.colostate.edu/helpdocs/vi.html
  - Vim Basics in 8 Minutes https://youtu.be/ggSyF1SVFr4

More awk, bash, regex, sed (1/9)

- awk
  - `awk '{print "Hello awk!"}' file.txt` — print "Hello awk!" for every file.txt line.
  - `awk '{print $0}' file.txt` — print every file.txt line.
  - `awk '{print $1}' file.txt` — print first field of every file.txt line.
  - `awk '{print $2}' file.txt` — print second field of every file.txt line.
- regex
  - to search patterns
  - BRE (Basic Regular Expression) vs ERE (Extended Regular Expression)
  - Flavors: Grep, Java, JavaScript, PHP, POSIX, Python, sed, XML, ...

More awk, bash, regex, sed (2/9)

- ≪`^$`≫ — matches a beginning-of-line + end-of-line (empty line).
  - ≪`^`≫ — matches a beginning-of-line (meaningless).
  - ≪`^hello$`≫ — matches just "hello" in a line.
- ≪`.`≫ — matches any character.
  - ≪`hell.`≫ — matches "hellA", "hella", "hellB", "hellb", ...
- ≪`[AB]`≫ — matches "A" or "B" only.
  - ≪`[0-3]`≫ — matches "0", "1", "2", or "3" only.
  - ≪`[^4-9]`≫ — not match "4", "5", "6", "7", "8", or "9".
- ≪`?`≫ — matches preceding zero or one time.
  - ≪`a?b`≫ — matches "b" or "ab" only.
- ≪`*`≫ — matches preceding zero or more times.
  - ≪`a*b`≫ — matches "b" or "ab" or "aab" or ...
  - ≪`A.*Z`≫ — matches "AZ" or "AaZ" or "AabZ" or ...
- ≪`+`≫ — matches preceding one or more times.
  - ≪`a+b`≫ — matches "ab", "aab", "aaab", ...

More awk, bash, regex, sed (3/9)

- ≪{}≫ — matches numbers in {}.
  - ≪a{2}≫ — matches "aa".
  - ≪a{2,5}≫ — matches "aa", "aaa", "aaaa", and "aaaaa".
  - ≪a{2,}≫ — matches "aa", "aaa", "aaaa", "aaaaa", ...
- ≪\≫ — escape character.
- ≪\0≫ — NULL.
- ≪\b≫ — word boundary.
- ≪\B≫ — non-word boundary.
- ≪\d≫ — any digit. E.g. ≪\d{1,3}≫ = 0 - 999.
- ≪\D≫ — any non-digit.
- ≪\n≫ — new line.
- ≪\t≫ — tab.
- ≪\s≫ — white space character.
- ≪\S≫ — non white space character.

More awk, bash, regex, sed (4/9)

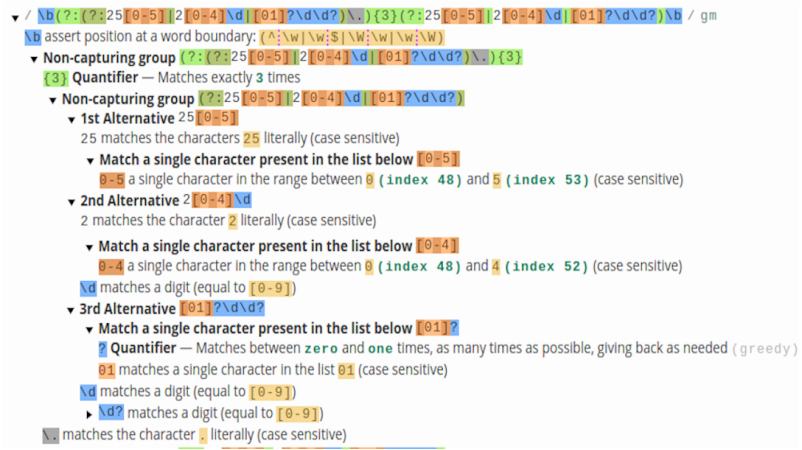
- ≪`(...)`≫ — group.
  - ≪`(?:...)`≫ — passive group.
  - ≪`(regex)|(regex)`≫ — matches left regex or right regex.
  - ≪`(a|b`≫ — matches either a or b.
  - ≪`^(From|To):`≫ — matches either ≪`^From:`≫ or ≪`^To:`≫.
- ≪`[0-9]{10}`≫ — 10 digits.
- ≪`0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]`≫ — 00:00–23:59.
- ≪`([0-9]|0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]`≫ — (0)0:00–23:59.

More awk, bash, regex, sed (5/9)

- ≪`[:alnum:]`≫ — alpha-numerics.
- ≪`[:alpha:]`≫ — alphabets
- ≪`[:blank:]`≫ — spaces and tabs.
- ≪`[:digit:]`≫ — digits.
- ≪`[:lower:]`≫ — lower case.
- ≪`[:space:]`≫ — spaces.
- ≪`[:upper:]`≫ — upper case.
- ≪`[:xdigit:]`≫ — hexadecimal digits.
- ≪`[:punct:]`≫ — punctuation.
- ≪`[:cntrl:]`≫ — control characters.
- ≪`[:graph:]`≫ — printed characters.
- ≪`[:print:]`≫ — printed and spaces.
- ≪`[:word:]`≫ — alpha-numerics and underscore.

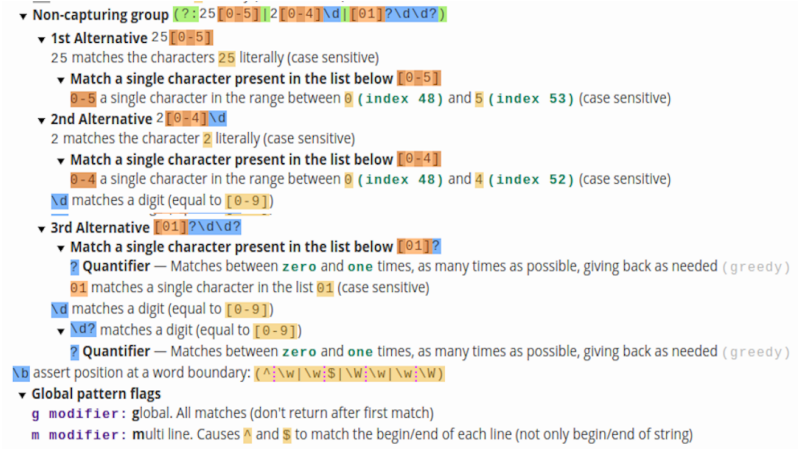
More awk, bash, regex, sed (6/9)

```
\b(?:(?:25[0-5]|2[0-4]\d|[01]?\d\d?)\.){3}
(?:25[0-5]|2[0-4]\d|[01]?\d\d?)\b
```



More awk, bash, regex, sed (7/9)

```
\b(?:(?:25[0-5]|2[0-4]\d|[01]?\d\d?)\.){3}
(?:25[0-5]|2[0-4]\d|[01]?\d\d?)\b
```



More awk, bash, regex, sed (8/9)

```
# file.txt

1. This is no. 1.
2. This is no. 22.
3. This is no. 333.
4. This is no. 4 4 4 4.
5. This is Joko.
6. This is Joko Joko.
7. This is joko.
8. This is Bowo.
9. This is bowo.
sed    'G'    ZA-thisfile1.txt
sed    'G;G'  ZA-thisfile1.txt
sed -n '4,6p' ZA-thisfile1.txt
sed -n '4,6p' ZA-thisfile1.txt > ZA-thisfile2.txt
sed -n '/[0-9]\{2\}/p' ZA-thisfile1.txt
sed    '4,6d' ZA-thisfile1.txt
sed    '$d' ZA-thisfile1.txt
sed    '5,/HABATS/d'   ZA-thisfile1.txt
sed    's/Joko/Bowo/'   ZA-thisfile1.txt
sed    's/Joko/Bowo/2' ZA-thisfile1.txt
sed    's/Joko/Bowo/g' ZA-thisfile1.txt
sed    's/Bowo\|bowo/Joko/g' ZA-thisfile1.txt
awk    '{print "Hello awk!"}' ZA-thisfile1.txt
awk    '{print $0}' ZA-thisfile1.txt
awk    '{print $1}' ZA-thisfile1.txt
awk    '{print $2}' ZA-thisfile1.txt
HABATS: This is the last line, dude!
```

More awk, bash, regex, sed (9/9)

- `sed 'G' file.txt` — double space.
- `sed 'G;G' file.txt` — triple space.
- `sed -n '4,6p' file.txt` — show only line 4 to 6.
- `sed -n '4,6p' file.txt > newfile.txt` — write line 4 to 6 to newfile.txt.
- `sed '/[0-9]\{2\}/p' file.txt` — show only lines with two digits.
- `sed '4,6d' file.txt` — show all except line 4 to 6.
- `sed '$d' file.txt` — show all except last line.
- `sed '5,/HABATS/d'` — show all except from line 5 to a line with HABATS.
- `sed 's/Joko/Bowo/' file.txt` — replace Joko with Bowo.
- `sed 's/Joko/Bowo/2' file.txt` — replace the second Joko with Bowo.
- `sed 's/Joko/Bowo/g' file.txt` — replace every Joko with Bowo.
- `sed 's/Bowo\|bowo/Joko/g' file.txt` – replace every Bowo or bowo with Joko.

The End

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