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

Rahmat M. Samik-Ibrahim (ed.)

University of Indonesia

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

REV263 10-Feb-2021

Operating Systems 211[3]) — **PJJ from HOME** ZOOM: A [Mon (or Wed) 10] — B [Mon (or Wed) 15] — C [Tue (or Thu) 08]

Week	**Schedule & Deadline**[1])	**Topic**	**OSC10**[2])
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.

[1]) The **DEADLINE** of Week 00 is 28 Feb 2021, whereas the **DEADLINE** of Week 01 is 07 Mar 2020, and so on...

[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 also http://codex.cs.yale.edu/avi/os-book/OS10/.
- **Resources**
  - **SCELE** — https://scele.cs.ui.ac.id/course/view.php?id=3020. 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).
- **Build your own Virtual Guest**
  https://osp4diss.vlsm.org/

Agenda

Agenda (2)

Week 01 Overview II: Topics[1]

- Types of virtualization (including Hardware/Software, OS, Server, Service, Network)
- Paging and virtual memory
- Virtual file systems
- Hypervisors
- Portable and cost of virtualization; emulation vs. isolation
- Cloud services: IAAS, PAAS and Platform APIs, SAAS
- Introduction to Scripting and REGEX.

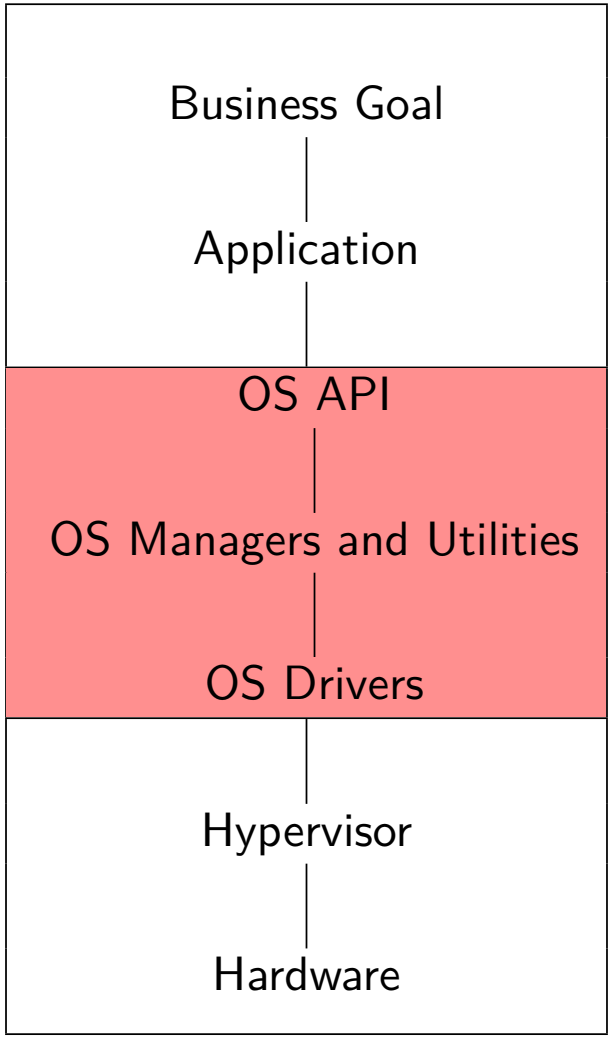
[1] Source: ACM IEEE CS Curricula 2013

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 virtualized infrastructure. [Familiarity]

[1] Source: ACM IEEE CS Curricula 2013

The Operating System



Week 01: Review 2 & Scripting

- Pengenalan Lisensi Perangkat Lunak Bebas: `https://rms46.vlsm.org/1/70.pdf`
- The Minix3 Notes: `https://rms46.vlsm.org/2/166.pdf`
- Intelectual Property Right (IPR)
- Operating System Services
- User Operating System Interface
- System Calls
- Types of System Calls
- System Programs
- Operating System Design and Implementation
- Operating System Structure

Intelectual Property Right (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.
- Paten *(Patent)* — UU no. 14/2001.
- Copyright (Hak Cipta) — UU no. 19/2002.
- The problem of Intelectual Property Right (IPR).
- Software IPR.
- Software Licenses: GNU GPL, EULA. Public Domain, Apache, Microsoft Public License.

Is this a Software Patent or Not?

The AV1 Video Codec — Timothy B. Terriberry — LINUX.CONF.AU 21-25 January 2019

EUROPEAN PATENT SPECIFICATION

(11) EP 0 460 751 B1

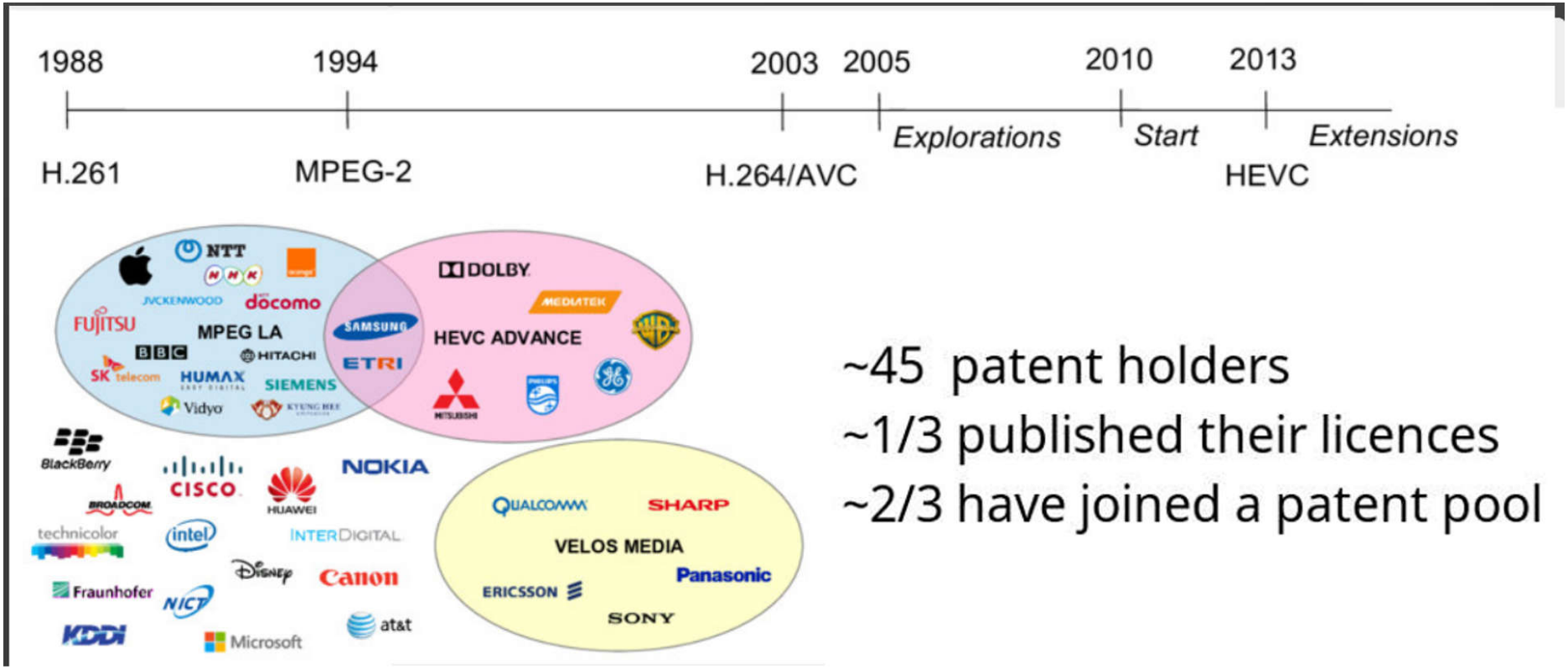
Date of filing: **03.06.1991**

Method of transmitting audio and/or video signals

Description

The invention relates to a method of transmitting audio and/or video signals *via* some transmission medium. More particularly the transmission medium is constituted by an optically readable disc. However, the transmission medium may also be a magnetic tape or disc or a direct connection between a transmitter and a receiver. The invention also relates to the transmission medium on which the audio and/or video signals are recorded, to an encoding apparatus for transmitting the audio and/or video signals, and to a decoding apparatus for receiving these signals.

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 so 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)

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.

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.

Week 01: Assignment #1

- Setting a Debian Guest `https://osp4diss.vlsm.org/`
  - Option 1: Create a Debian Virtual Guest from Scratch.
    - FDM: Free Download Manager (Optional)
      `https://www.freedownloadmanager.org/`
    - Downloading Debian Netinst
      `https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/`
    - Downloading and Installing VirtualBox
      urlhttps://www.virtualbox.org/
    - Installing Debian NetInst (guest) on VirtualBox
      `https://osp4diss.vlsm.org/InstallDebianNetinst.html`
    - More Debian Packages
      `https://osp4diss.vlsm.org/MoreDebianPackages.html`
  - Option 2: Download an OVA File
    - For a limited time only!
    - README: `https://bit.ly/3mxkpvP` (182 bytes)
    - Debian 10.5 OVA for VirtualBox: `https://bit.ly/2FMU7F8` (662MB)

Week 01: Assignment #2 (1): Some Essential Commands

- Read by **GSGS**[1]
  - Machtelt Garrels: Bash Guide for Beginners.
  - Mendel Cooper: An in-depth exploration of the art of shell scripting — Advanced Bash-Scripting Guide.
  - Jan Goyvaerts: Regular Expressions — The Complete Tutorial.
- The **ATM Way**[2].
  - Setup a GNU/Linux Guest on VirtualBox. See also `osp4diss.vlsm.org/`.
  - *Clone* Demo from `https://github.com/UI-FASILKOM-OS/SistemOperasi.git`
  - Learn **login** and **logout** with `ssh` or `putty`.
  - Pick an editor, eg. (**vi**).
- Learn some **Command-Line Interface** (CLI) commands.
  - shell (Bash)
  - basic CLI: cat, cd, cp, ls, man, more, mv, rm, touch, wc.
  - vi, sed, awk, git.

[1] Google Sana, Google Sini

[2] Amati, Tiru, Modifikasi. Romi Satria Wahono has been using this term since 2007.

Week 01: Assignment #2 (2): Some Essential Commands

man	manual. Eg. "man man"
passwd	changes passwords.
ls	list directory contents. Eg. "ls -al"
cd	change the working directory. Eg. "cd /tmp"
cp	copy file(s). Eg. "cp SOURCE DEST"
rm	remove file(s). Eg. "rm AFILE"
mv	move files(s). Eg. "mv FROMFILE TOFILE"
mkdir	make directories(s). Eg. "mkdir ADIRECTORY"
rmdir	remove directories(s). Eg. "rmdir ADIRECTORY"
cat	read file(s) Eg. "cat AFILE"
more	read file(s) per screen Eg. "more AFILE"
ln	make a link of a file. Eg. "ln -s file sfile"
grep	search string aword inside file. Eg. "grep aworld file"
sort	sort lines of text files. Eg. "sort file1.txt"
top	display systems task. Eg. "top"
find	Eg. "find / -name minix3.iso -print". Find from "/".

Week 01: Assignment #2 (3): Some Essential Commands

chmod	Eg. `"chmod 755 file"`. Change file with access mode 755.
chown	Eg. `"chown user file"`. Change owner file to user.
chgrp	Eg. `"chgrp other file"`. Change group file to other.
tar	tape archive file. Eg.
	`"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. Eg. `"date +%Y"`
tee	read from standard input and write to standard output and files.
	Eg. `"ls -al | tee listing.txt"`
diff	compare files line by line. Eg. `"diff file1.txt file2.txt"`
wc	print newline, word, and byte counts for each file.
	Eg. `"wc file.txt"`

Week 01: Assignment #2 (4): The "vi" editor

- VI Basics

	Basics		More Commands
i	insert mode	d^	delete from ^ (beginning) to the curs
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`
  - How to Use the vi Editor
    `https://www.washington.edu/computing/unix/vi.html`
  - Vim Basics in 8 Minutes `https://youtu.be/ggSyF1SVFr4`

Week 01: Assignment #2 (5): REGEX

- to search patterns
- BRE (Basic Regular Expression) vs ERE (Extended Regular Expression)
- Flavors: Grep, Java, JavaScript, PHP, POSIX, Python, sed, XML, ...

Week 01: Assignment #2 (6): More REGEX

- ≪`^$`≫ — matches a beginning-of-line + end-of-line (empty line).
  - ≪`^`≫ — matches a beginning-of-line (meaningless).
  - ≪`^hello$`≫ — 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", ...
- ≪`{}`≫ — matches numbers in {}.
  - ≪`a{2}`≫ — matches "aa".
  - ≪`a{2,5}`≫ — matches "aa", "aaa", "aaaa", and "aaaaa".
  - ≪`a{2,}`≫ — matches "aa", "aaa", "aaaa", "aaaaa", ...

Week 01: Assignment #2 (7): More REGEX

- ≪\≫ — escape character.
- ≪\0≫ — NULL.
- ≪\b≫ — word boundary.
- ≪\B≫ — non-word boundary.
- ≪\d≫ — any digit. Eg. ≪\d{1,3}≫ = 0 - 999.
- ≪\D≫ — any non-digit.
- ≪\n≫ — new line.
- ≪\t≫ — tab.
- ≪\s≫ — white space character.
- ≪\S≫ — non white space character.

Week 01: Assignment #2 (8): More REGEX

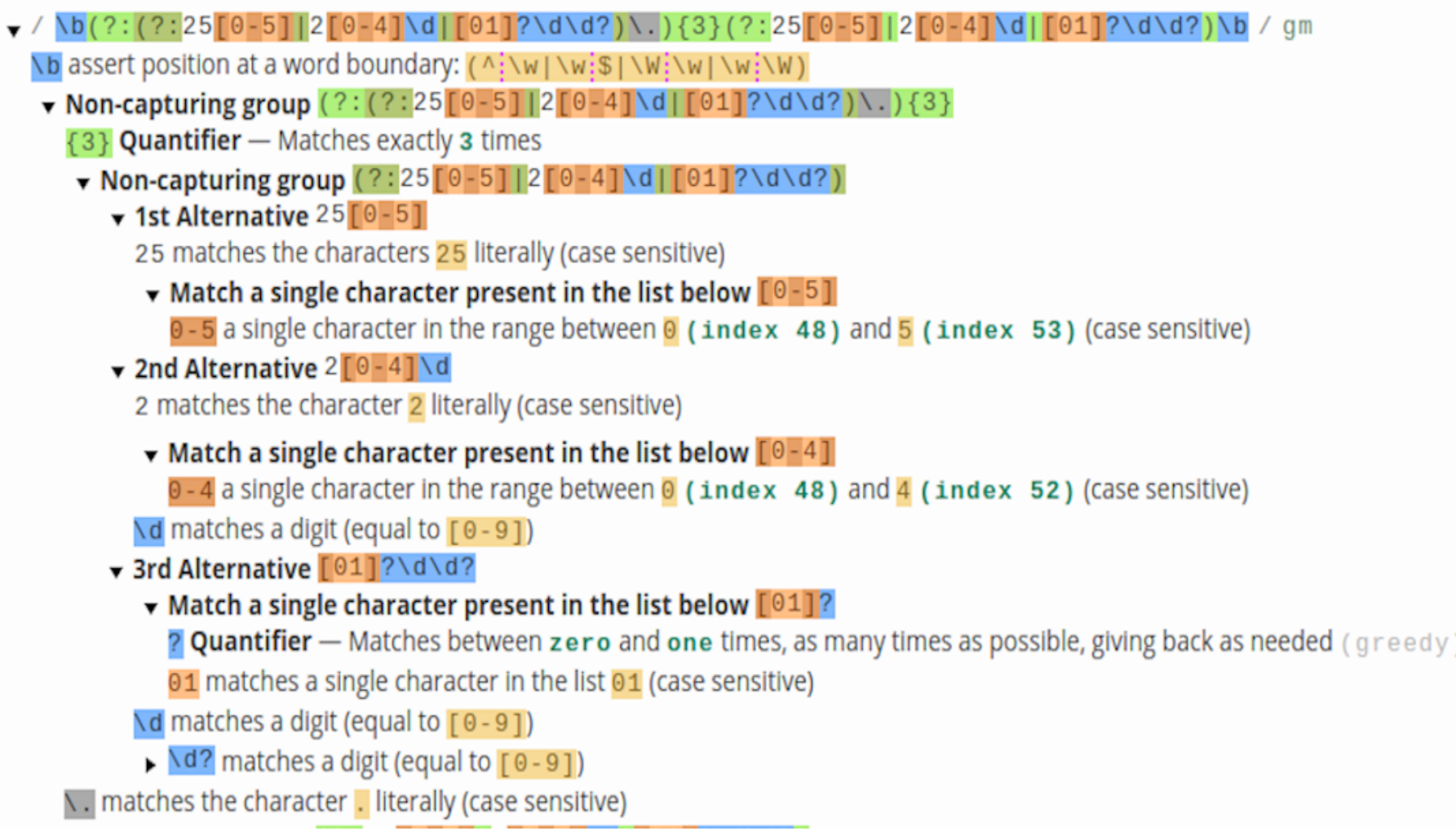
- ≪`(...)`≫ — group.
  - ≪`(?:...)`≫ — pasive group.
  - ≪`(regex)|(regex)`≫ — matches left regex or right regex.
  - ≪`(a|b`≫ — matches either a or b.
  - ≪`^(From|To):`≫ — matches either ≪`^From:`≫ or ≪`^To:`≫.
- ≪`[0-9]{10}`≫ — 10 digits.
- ≪`0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]`≫ — 00:00–23:59.
- ≪`([0-9]|0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]`≫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

Week 01: Assignment #2 (9): More REGEX

- ≪`[: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.

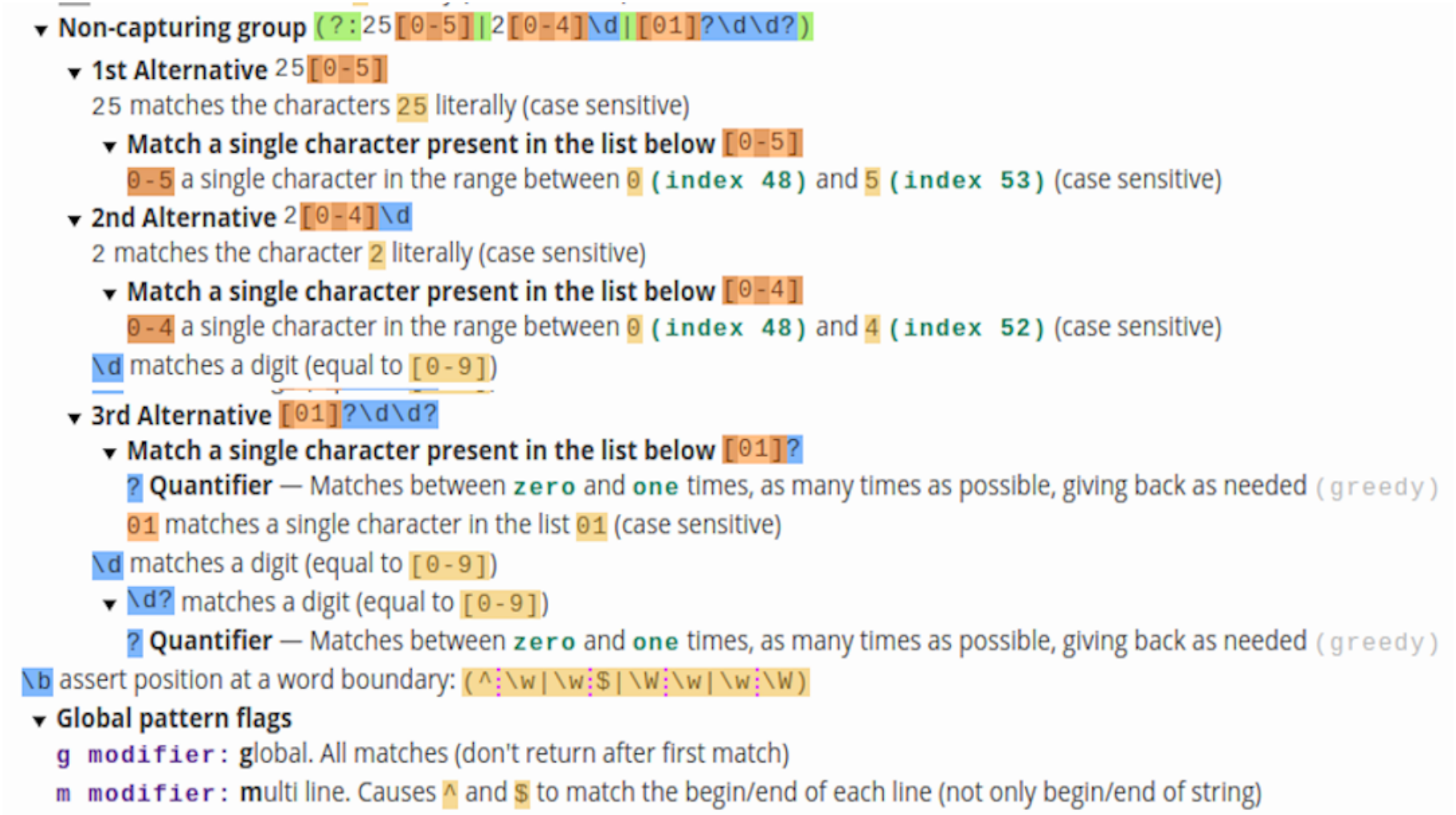
Week 01: Assignment #2 (10): Regex101.com

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



Week 01: Assignment #2 (11): Regex101.com

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



Week 01: Assignment #2 (12): stream editor(sed)

- `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.

Week 01: Assignment #2 (13): 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.

Week 01: Assignment #2 (14): Cloning GITHUB

```
cbkadal@osp:~$ PS1=">>>>> $ "

>>>>> $ git clone https://github.com/UI-FASILKOM-OS/SistemOperasi.git
Cloning into 'SistemOperasi'...
remote: Enumerating objects: 51, done.
remote: Counting objects: 100% (51/51), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 972 (delta 29), reused 34 (delta 27), pack-reused 921
Receiving objects: 100% (972/972), 24.63 MiB | 4.65 MiB/s, done.
Resolving deltas: 100% (637/637), done.

>>>>> $ ls -F SistemOperasi/
CNAME  _config.yml  demos/  LICENSE  OLDREADME.md  pdf/  README.md

>>>>> $ ls -al SistemOperasi/demos/
total 56
drwxr-xr-x 14 demo demo 4096 Jan 16 14:24 .
drwxr-xr-x  5 demo demo 4096 Jan 16 14:24 ..
drwxr-xr-x  2 demo demo 4096 Jan 16 14:24 Week00
drwxr-xr-x  2 demo demo 4096 Jan 16 14:24 Week01
drwxr-xr-x  4 demo demo 4096 Jan 16 14:24 Week02
drwxr-xr-x  2 demo demo 4096 Jan 16 14:24 Week03
drwxr-xr-x  2 demo demo 4096 Jan 16 14:24 Week04
drwxr-xr-x  2 demo demo 4096 Jan 16 14:24 Week05
drwxr-xr-x  2 demo demo 4096 Jan 16 14:24 Week06
drwxr-xr-x  2 demo demo 4096 Jan 16 14:24 Week07
drwxr-xr-x  2 demo demo 4096 Jan 16 14:24 Week08
drwxr-xr-x  4 demo demo 4096 Jan 16 14:24 Week09
drwxr-xr-x  2 demo demo 4096 Jan 16 14:24 Week10
drwxr-xr-x  2 demo demo 4096 Jan 16 14:24 WeekTMP
>>>>> $
```

Inside the "week01-scripting" folder

```
>>>>> $ pwd
/home/cbkadal/mydemo/W01-demos

>>>>> $ ls -al
total 96
drwxr-xr-x  2 demo demo 4096 Jan 23 18:38 .
drwx------ 14 demo demo 4096 Jan 23 18:38 ..
-rw-r--r--  1 demo demo 1797 Jan 23 18:38 1-READ-THIS-FIRST.txt
-rw-r--r--  1 demo demo 4880 Jan 23 18:38 a01-READ-ME
-rw-r--r--  1 demo demo 5644 Jan 23 18:38 a02-sort-n-prepare
-rw-r--r--  1 demo demo 4644 Jan 23 18:38 a03-command-lines-demo
-rw-r--r--  1 demo demo 1193 Jan 23 18:38 a04-does-it-exist
-rw-r--r--  1 demo demo 1204 Jan 23 18:38 a05-finding-EXIST
-rw-r--r--  1 demo demo 1114 Jan 23 18:38 a06-loop
-rw-r--r--  1 demo demo 1518 Jan 23 18:38 a07-tester
-rw-r--r--  1 demo demo 1577 Jan 23 18:38 a08-append-a-file
-rw-r--r--  1 demo demo 1168 Jan 23 18:38 a09-add-numbers
-rw-r--r--  1 demo demo 1569 Jan 23 18:38 a10-mysha1
-rw-r--r--  1 demo demo 2271 Jan 23 18:38 a11-banding
-rw-r--r--  1 demo demo 2110 Jan 23 18:38 a12-fixfs
-rw-r--r--  1 demo demo 1576 Jan 23 18:38 a13-last
-rw-r--r--  1 demo demo  752 Jan 23 18:38 a14-absen
-rw-r--r--  1 demo demo 1187 Jan 23 18:38 a15-uts171
-rw-r--r--  1 demo demo  522 Jan 23 18:38 a16-uts181
-rw-r--r--  1 demo demo  536 Jan 23 18:38 a17-uts182
-rw-r--r--  1 demo demo  404 Jan 23 18:38 .head
>>>>> $
```

Demo Files(1)

- 000-READ-THIS-FIRST.txt
- a01-SCREEN-CHECK: if the screen is at least 80 x 23.
- a02-sort-n-prepare: folder sorting; preparing and deleting folders.
- a03-command-lines-demo: CLI demo.
- a04-does-it-exist
- a05-finding-EXIST
- a06-loop
- a07-tester
- a08-append-a-file
- a09-add-numbers
- a10-mysha1

Demo Files(2)

- a11-banding
- a12-fixfs
- a13-last
- a14-absen
- a15-uts171
- a16-uts181
- a17-uts182
- a18-uts191
- a19-uts192
- a20-uts201

Week 01: Assignment #2 (13) Some URLs

- Try some CLI commands
  `https://osp4diss.vlsm.org/Welcome2GNULinux.html`
- Introduction to Linux and Basic Linux Commands for Beginners – `https://youtu.be/IVquJh3DXUA`
- The Complete Linux Course: Beginner to Power User (7:23 hours) – `https://youtu.be/wBp0Rb-ZJak`
- The Linux command line for beginner – `https://ubuntu.com/tutorials/command-line-for-beginners`
- 40 Basic Linux Commands used Frequently – `https://linoxide.com/linux-command/essential-linux-basic-commands/`
- Regular Expression (REGEX) Tester – `https://regex101.com/`
- Regex Tutorial — A Quick Cheatsheet by Examples
  `https://medium.com/factory-mind/regex-tutorial-a-simple-cheatsheet-by-examples-649dc1c3f2`

Week 01: Assignment #3 - #8

- See https://osp4diss.vlsm.org/MoreGNULinux.html
  - Create a new user account such as your GitHub account (eg. "cbkadal").
- See https://osp4diss.vlsm.org/CBKadal.html
  - Create a tunnel from your guest to badak.cs.ui.ac.id via kawung.cs.ui.ac.id.
  - Copy folders from (rsync) badak:///extra/Doc/, badak:///extra/Slides/, and badak:///extra/Demos/,
  - GIT PULL your "os211" repository from GitHub.com.
  - Update and PUSH back your log mylog.txt.
- See https://github.com/cbkadal/os211/
  - Create two files w00.md and w01.md for your weekly TOP 10 list.
  - Compare (W00): https://raw.githubusercontent.com/cbkadal/os211/master/w00.md and https://cbkadal.github.io/os211/W00/.
  - Compare (W01): https://raw.githubusercontent.com/cbkadal/os211/master/w01.md and https://cbkadal.github.io/os211/W01/.

Week 01: Check List (Deadline: Monday, 28-Sep-2020).

- **Starting Point:** `https://os.vlsm.org/`
- Week 01: Assignment (more details in **os01.pdf**).
  1. **Create or Import** a Debian Virtual Guest (e.g. hostname "`osp`").
  2. Log into the guest and learn some CLI commands (e.g. `vi` editor).
  3. Create a new user account such as your GitHub account (e.g. "`cbkadal`").
  4. Create a tunnel from your guest to `badak.cs.ui.ac.id` via kawung.cs.ui.ac.id.
  5. Copy folders from (`rsync`) `badak:///extra/Doc/`, `badak:///extra/Slides/`, and `badak:///extra/Demos/`,
  6. GIT PULL your "`os211`" repository from `GitHub.com`.
  7. Update and PUSH back your log `mylog.txt`.
  8. Create two files `w00.md` and `w01.md` for your weekly TOP 10 list. For example, see `https://github.com/cbkadal/os211/` and `https://cbkadal.github.io/os211/`.
  9. Read: (OSC10 chapter 1 + chapter 2 + chapter 18)
- The "Assignment Day" is every Thursday morning.
- This page is `https://os.vlsm.org/Slides/check01.pdf`.

The End

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