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!

REV219 05-Feb-2020

Operating Systems 2020-1

A [08-10, Rm 3114, Mo/We] — B/M [10:10-12, Rm 3114, Mo/We] — C [13-15, Rm 3114, Mo/We]

D [10-12, Rm 2307(Mo), Rm 3113(We)] — E [08-10, Rm 2307(Mo), Rm 3113(We)]

Week	Schedule	Topic	OSC10
Week 00	27 Jan - 02 Feb 2020	Overview 1, Virtualization & Scripting	Ch. 1, 2, 18.
Week 01	03 Feb - 09 Feb 2020	Overview 2, Virtualization & Scripting	Ch. 1, 2, 18.
Week 02	10 Feb - 16 Feb 2020	Security, Protection, Privacy, & C-language	Ch. 16, 17
Week 03	17 Feb - 23 Feb 2020	File System & FUSE	Ch. 13, 14, 15
Week 04	24 Feb - 01 Mar 2020	Addressing, Shared Lib, & Pointer	Ch. 9
Week 05	02 Mar - 08 Mar 2020	Virtual Memory	Ch. 10
Reserved	09 Mar - 13 Mar 2020	Q & E	
MidTerm	14-21 Mar 2020 (TBA)	MidTerm (UTS)	Subject to change.
Week 06	23 Mar - 31 Mar 2020	Concurrency: Processes & Threads	Ch. 3, 4
Week 07	01 Apr - 07 Apr 2020	Synchronization & Deadlock	Ch. 6, 7, 8
Week 08	08 Apr - 14 Apr 2020	Scheduling + W06/W07	Ch. 5
Week 09a	15 Apr - 19 Apr 2020	Storage, Firmware, Bootldr, & Systemd	Ch. 11
OCL	20 Apr - 26 Apr 2020	OnLine & CoLearnIng	
Week 09b	27 Apr - 28 Apr 2020	Storage, Firmware, Bootldr, & Systemd	Ch. 11
Week 10	29 Apr - 05 May 2020	I/O & Programming	Ch. 12
Reserved	06 May - 10 May 2020	Q & A	
Final	11-18 May 2020 (TBA)	Final (UAS)	This schedule is subject to change.
Extra	25 Jun 2020	Extra assignment confirmation	

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**
  - **All In One** — BADAK.cs.ui.ac.id:///extra/ (**FASILKOM only!**).
  - **Download Slides and Demos from GitHub.com** https://github.com/UI-FASILKOM-OS/SistemOperasi/
  - **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).
- **Try Demos**
  - Your own Ubuntu system.
  - Ubuntu on VirtualBox, or VMWare, or ...
  - Windows Subsystem for Linux (**Windows 10 only!**).
  - SSH to BADAK.cs.ui.ac.id (**FASILKOM only!**).

Agenda

Agenda (2)

11. Regex: Regular Expressions
12. vi
13. sed
14. awk
15. Demo
16. Week 01: Summary
17. Week 01: Check List
18. The End

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

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`
- Linux Help: `https://www.mediacollege.com/linux/`
- 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)

- Rahasia Dagang *(Trade Sceret)* — UU no. 30/2000.
- Desain Industri *(Industrial Design)* — UU no. 31/2000.
- Desain Tata Letak Sirkuit Terpadu *(Integrated Circuit Layout Design)* — UU no. 32/2000.
- Paten *(Patent)* — UU no. 14/2001.
- Hak Cipta *(Copyright)* — UU no. 19/2002.
- Konsekuensi HKI
- HKI Perangkat Lunak
- Lisensi Perangkat Lunak: 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

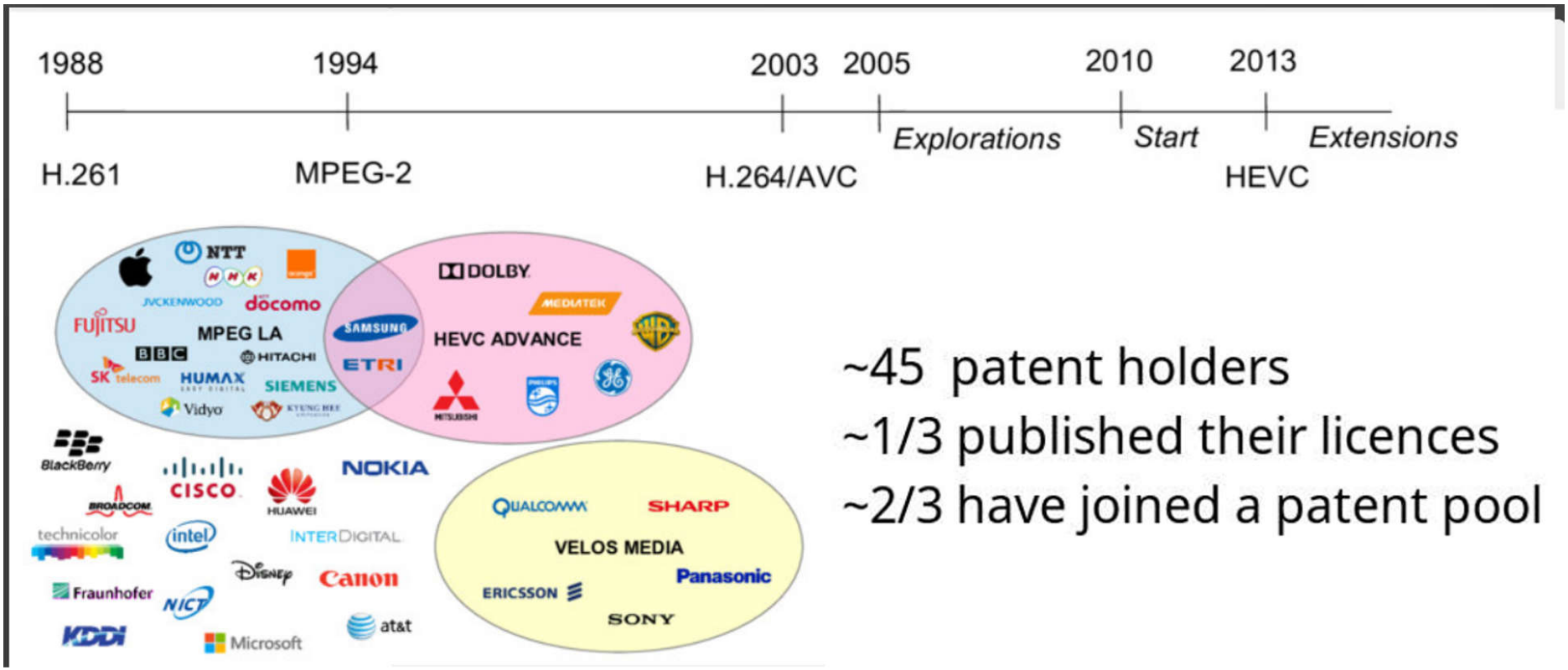
Date of filing: **03.06.1991**

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

FOUNDING MEMBERS

amazon, Apple, arm, cisco, facebook, Google, IBM, intel, Microsoft, moz://a, NETFLIX, NVIDIA

PROMOTER MEMBERS

Adobe, Alibaba Group, ALLEGRO, AMD, Videolan, vimeo, Amlogic, ARGON DESIGN, Ateme, BBC R&D, 微帧 Visionular, XILINX, 金山云, BITMOVIN, BROADCOM, CableLabs, Chips Media, gfycat, hulu, Ittiam, iQIYI 爱奇艺, NGCODEC, Polycom, REALTEK, socionext, V-SILICON, VeriSilicon, Vidyo

Source (per 18-Feb-19): 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.

Scripting

- Readings (do Google!)
  - 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 (Amati, Tiru, Modifikasi)[1].
  - *Clone* Demo
    `https://github.com/UI-FASILKOM-OS/SistemOperasi.git`
  - **GSGS — ATM**: Google Sana, Google Sini: Amati, Tiru, Modifikasi!
  - Medium: `badak.cs.ui.ac.id`
  - Opsi: BYOD, WSL (Windows 10), CYGWIN.
  - Belajar **login** dan **logout** dengan `ssh` atau `putty`[2].
  - Belajar editor yang bagus punya buatan (**vi**).
- Belajar beberapa perintah **Command-Line Interface** (CLI).
  - shell (Bash)
  - basic CLI: cat, cd, cp, ls, man, more, mv, rm, touch, wc.
  - vi, sed, awk, git.

[1] Romi Satria Wahono sudah menggunakan istilah ini sejak tahun 2007 (Google).

[2] Sesuai dengan keyakinan dan kepercayaan masing-maing.

External Resources

- Linux Resources:
  `http://www.mediacollege.com/linux/`
- Tutorial:
  `https://www.mediacollege.com/linux/command-tutorial/`
- Commands:
  `https://www.mediacollege.com/linux/command/linux-command.html`
- Shell:
  `https://www.mediacollege.com/linux/command/shell-command.html`
- Regular Expression (REGEX):
- REGEX Tester:
  `https://regex101.com/`

Some Essential Command Line Commands part 1

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 "/".

Some Essential Command Line Commands part 2

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"`

Regex: Regular Expressions

- ≪`^$`≫ — matches a beginning-of-line + end-of-line (empty line).
  - ≪`^`≫ — matches a beginning-of-line (meaningless).
  - ≪`^hello$`≫ — 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", ...

More Regex 1

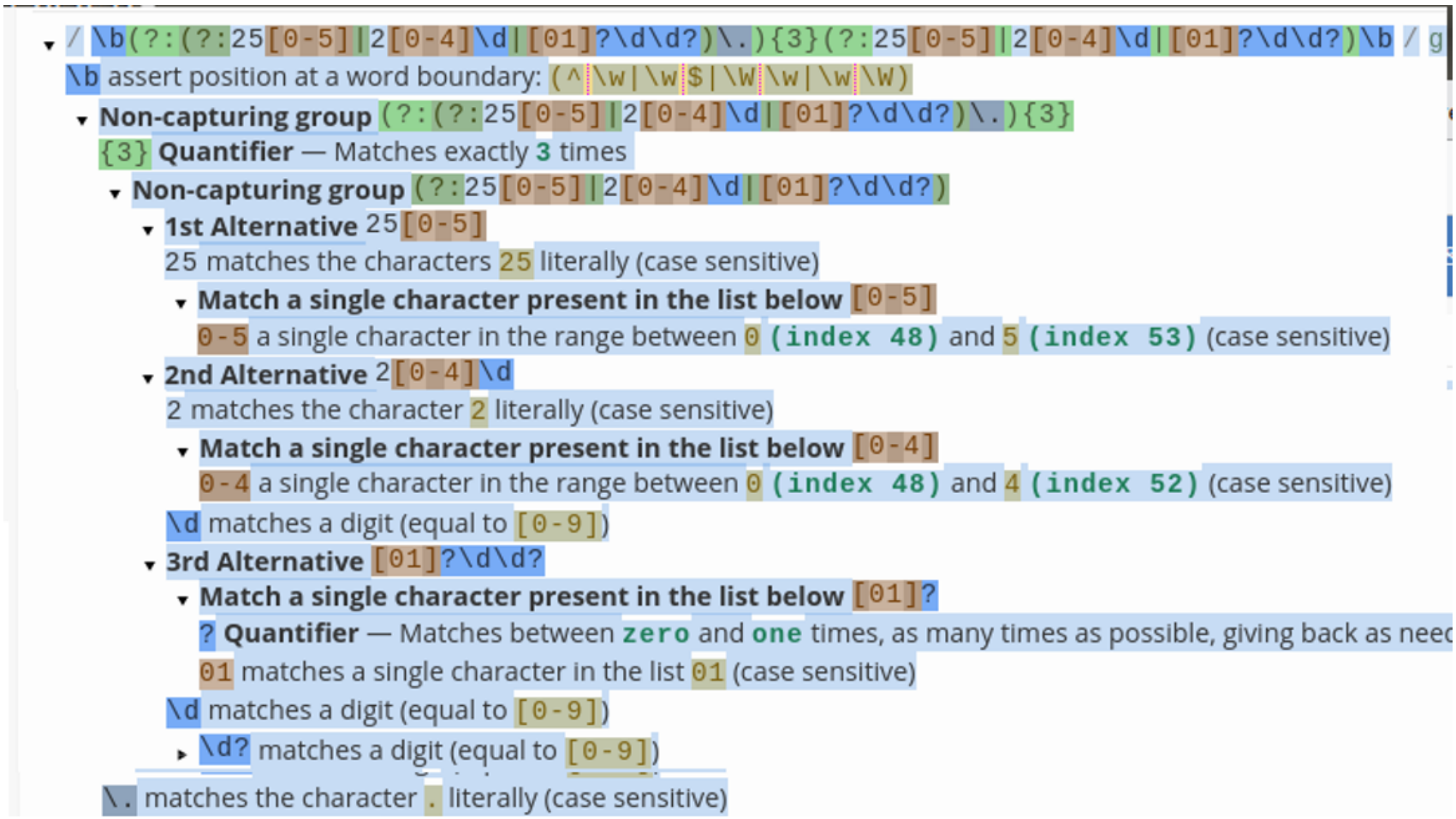
- ≪\≫ — escape character.
- ≪\0≫ — NULL.
- ≪\b≫ — word boundary.
- ≪\B≫ — non-word boundary.
- ≪\d≫ — any digit. Eg. ≪\d{1,3}≫ = 0 - 999.
- ≪\D≫ — any non-digit.
- ≪\n≫ — new line.
- ≪\t≫ — tab.
- ≪\s≫ — white space character.
- ≪\S≫ — non white space character.

More Regex 2

- ≪`[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.
- ≪`(regex)|(regex)`≫ — matches left regex or right regex.
  - ≪`(a|b`≫ — matches either a or b.
  - ≪`^(From|To):`≫ — matches either ≪`^From:`≫ or ≪`^To:`≫.

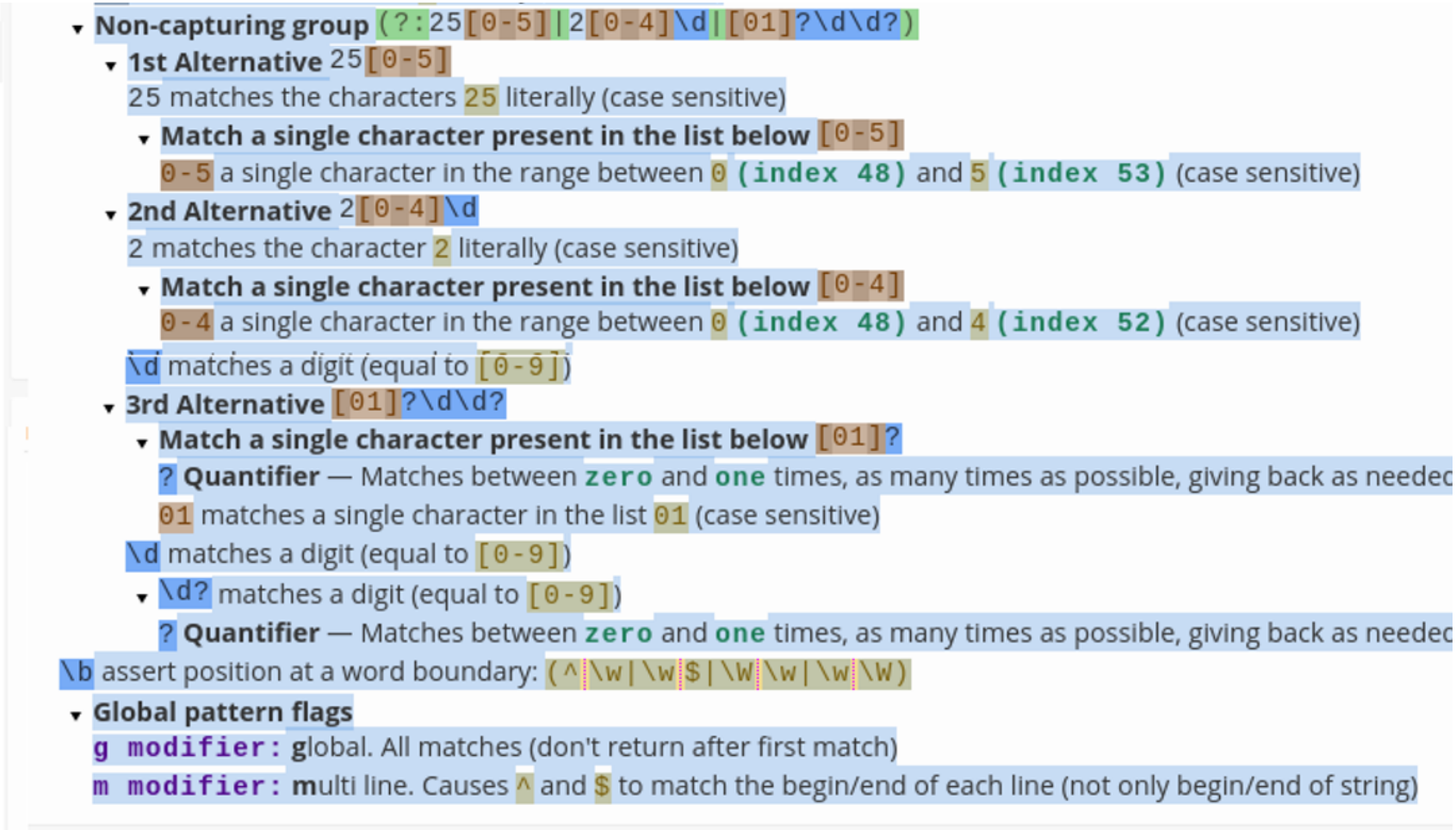
Regex101.com: IPv4 Address

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



Regex101.com (cont)

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



The "vi" editor

	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

sed — the stream editor

- `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.

awk — Aho - Weinberger - Kernighan

- `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.

Login into BADAK.cs.ui.ac.id

```
demo@badak:~$ 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/demo/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)

- 1-READ-THIS-FIRST.txt
- a01-READ-ME: Yes, read that!
- a02-sort-n-prepare: folder sorting; preparing and deleting folder ".ZTEST".
- a03-command-lines-demo: demo beberapa perintah CLI.
- a04-does-it-exist
- a05-finding-EXIST
- a06-loop
- a07-tester

Demo Files(2)

- a08-append-a-file
- a09-add-numbers
- a10-mysha1
- a11-banding
- a12-fixfs
- a13-last
- a14-absen
- a15-uts171
- a16-uts181
- a17-uts182

Week 01: Summary

- Reference: (OSC10 chapter 1 + chapter 2 + chapter 18)

Week 01: Check List

- [ ] **How to improve this document?**

The End

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