CSF2600505 Sistem Operasi
CSGE602055 Operating Systems
Minggu 00: Intro & Review

Rahmat M. Samik-Ibrahim

Universitas Indonesia

http://rms46.vlsm.org/2/207.html

REV70 13-Sep-2017

Jadwal OS172

Minggu 00	29 Aug - 05 Sep 2017	Intro & Review
Minggu 01	07 Sep - 12 Sep 2017	IPR, SED, AWK, REGEX, & Scripting
Minggu 02	14 Sep - 19 Sep 2017	Protection, Security, Privacy, & C-language
Minggu 03	26 Sep - 30 Sep 2017	BIOS, Loader, Systemd, & I/O
Minggu 04	03 Okt - 07 Okt 2017	Addressing, Shared Lib, Pointer & I/O Programming
Minggu 05	10 Okt - 14 Okt 2017	Virtual Memory
Ming. UTS	15 Okt - 24 Okt 2017	
Minggu 06	26 Okt - 31 Okt 2017	Concurency: Processes & Threads
Minggu 07	02 Nov - 07 Nov 2017	Synchronization
Minggu 08	09 Nov - 14 Nov 2017	Scheduling & Network Sockets Programming
Minggu 09	16 Nov - 21 Nov 2017	File System & Persistent Storage
Minggu 10	23 Nov - 28 Nov 2017	Special Topic: Blockchain
Cadangan	30 Nov - 09 Des 2017	
Ming. UAS	10 Des - 23 Des 2017	

Agenda

1. Start
2. Agenda
3. Pendahuluan
4. Goal
5. Assessment
6. Resources
7. Schedule
8. Self Service Lab
9. Memo Mingguan / QR Code
10. Tools
11. Lab Programing
12. Week 00: Introduction
13. Review
14. Managers Set
15. Potpourri
16. The End

Rahmat M. Samik-Ibrahim

- Bekerja di Universitas Indonesia: sejak 1984[1].
- Pengguna GNU/Linux: sejak 1994.
- VauLSMorg (vlsm.org): sejak 1996.
- Blog: `rahmatm.samik-ibrahim.vlsm.org/`
  - Blog: `2016/08/panggil-saya-rahmat.html`
  - Blog: `2013/10/kumpulan-hal.html`
  - Blog: `2011/08/ibu-ke-pasar-membeli-ayam.html`
- Twitter: `@rms46`
- Facebook (page only): `facebook.com/RMS46F/`
- Opsi Pembentukan OS172 Group?!
  Silakan Kontak via WhatsApp +62-881-456-**XXXX**

[1] MDCCXXXIV — Universitas Goettingen didirikan: **1734**

TOP 10 REALITA

1. Nama saya **Rahmat**. Rahmat nama saya. Kalau bukan Rahmat, bukan nama saya!
2. Jangan datang lebih lambat dari pada Pengajar! Terdapat dua kesempatan untuk menyusul masuk kelas: T+15 menit dan setelah istirahat.
3. Jangan berisik/asyik sendiri dalam kelas dan jangan main "*games*" dan "*chat*" dengan "*gadget*" anda!
4. Jangan lupa mempersiapkan diri untuk berpartisipasi dalam kelas!
5. Jangan lupa membawa selembar kertas (+QRC) untuk membuat memo kuliah!
6. Memo kuliah (+QRC) tersebut yang boleh dibawa saat UTS dan UAS.
7. Jangan curang!
8. Jangan meminjam peralatan selama kuis dan ujian!
9. Jangan menghubungi Pengajar untuk masalah Administratip!
10. Jangan menjadi "*Puss in Boot*"!

Jangan menjadi Puss In Boot

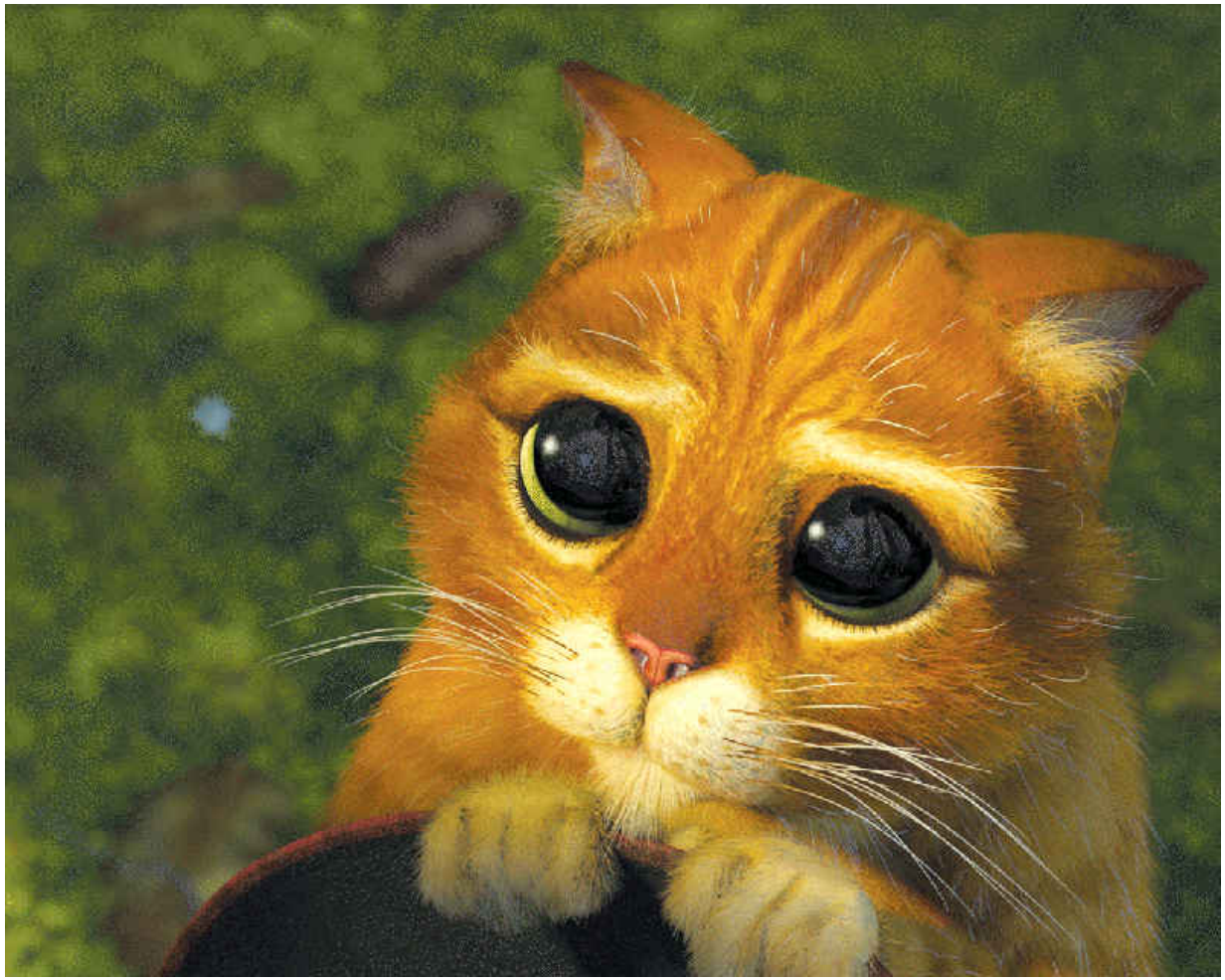


Figure: Ini Puss in Boot[1].

[1] This is a fair use of a DreamWorks/Paramount Picture character.

Goal

Coverage

This is an introduction to a modern operating systems course. It will cover general overview, computer architecture review, operating system overview, software licenses, GNU/Linux CLI, versioning, scripting, C language overview, protection, security, gnupg, processes and threads, addressing and pointers, memory management, virtual memory, synchronization, mutual exclusion, deadlock, CPU scheduling algorithms, file systems.

Student-Centered

This course is student-centered where responsibility is in the hands of the students. Students are expected to be prepared for the class meeting.

GNU/Linux

Students will have a thorough understanding of how GNU/Linux provides services by using a Command Line Interface.

Assessment

85 - ... = A	80 - 85 = A-	75 - 80 = B+	70 - 75 = B
65 - 70 = B-	60 - 65 = C+	55 - 60 = C	50 - 55 = D or C[1]
40 - 50 = D	30 - 40 = E	20 - 30 = E	00 - 20 = E

- **4 SKS**: Alokasikan 12 jam per minggu.
- **No Lab — No Task — No Assistant**.
- **Active Preparation Only — Active Participation Only**.
- UTS: 6 set problem @ 6 point (=36%).
- UAS: 5 set problem @ 6 point (=30%).
- Partisipasi Pra-UTS: 6 set @ 3 poin (=18%).
- Partisipasi Pasca-UTS: 5 set @ 3 point (=15%).
- Extra untuk nilai C keatas: 1 poin[1].
- C-2C untuk nilai C-: hingga 5 poin[1].

[1] Syarat dan Ketentuan Berlaku

Resources

- Silakan memilih buku Sistem Operasi yang terbit dalam 10 tahun terakhir.
- OLD — (ARSIP)(017_BAHAN-AJAR-LAMA) Previous Slides.
- SUP — (ARSIP)(041_Suplemen) Supplement.
- OSCE2e — (ARSIP)(050_OSC-Silberschatz) OSCE2e
- UCB — (ARSIP)(070_KULIAH-INTERNASIONAL) UC Berkeley
- UDA — (ARSIP)(070_KULIAH-INTERNASIONAL) UDACITY
- ETC — (ARSIP)(075_ETC-Video) ETC
- DEMO — (GITHUB) **https://github.com/UI-FASILKOM-OS/demo**
- SLIDE — (SLIDE) **http://rms46.vlsm.org/2/207.html**
- SCELE: **https://scele.cs.ui.ac.id/course/view.php?id=124**
  - Enrollment key: "01111000"[1]
- ARCHIVE (Arsip bahan pengajaran): **https://scele.cs.ui.ac.id/course/view.php?id=126**
  - Enrollment key: "11100001"[1].

[1] Sewaktu-waktu akan diganti!

Schedule part 1

- Week00 – Intro & Review (OSCE2e ch1/2)(UCB 01)(UDA P1L1/2) (OLD 00)
- Week01 – IPR, SED, AWK, REGEX, & Scripting (ETC 000 001 002)(OLD 02-HKI 02-scripting) (Any Related Tutorial)
- Week02 – Protection, Security, Privacy, & C-language (OSCE2e ch13-4) (ETC 050-1 C001-8) (OLD 01) (Any C Language Tutorial)
- Week03 – BIOS, Loader, Systemd, & I/O (Any Related Tutorial) (ETC 300-324) (SUP WEEK03)
- Week04 – Addressing, Shared Lib, Pointer & I/O Programming (OLD 08 10)
- Week05 – Virtual Memory (OSCE2e ch7/8) (UCB 11 12 13) (UDA P3L2) (OLD 06)
- UTS 00 01 02 03 04 05

Schedule part 2

- Week06 – Concurency: Processes & Threads (OSCE2e ch3/4) (UCB 02 03) (UDA P2L1/2/3) (OLD 03)
- Week07 – Synchronization (OSCE2e ch5) (UCB 7/8) (UDA P3L3/4) (OLD 04)
- Week08 – Scheduling & Network Sockets Programming (OSCE2e ch6) (UCB 9/10) (UDA P3L1) (OLD 05)
- Week09 – File System & Persistent Storage (OSCE2e ch9/10/11) (UCB 17A/18/19) (UDA P4L2 P4L2) (OLD 07 09) (SUP WEEK09)
- Week10 – Special Topic: Blockchain (TBA)
- UAS 06 07 08 09 10

Arsip Scele

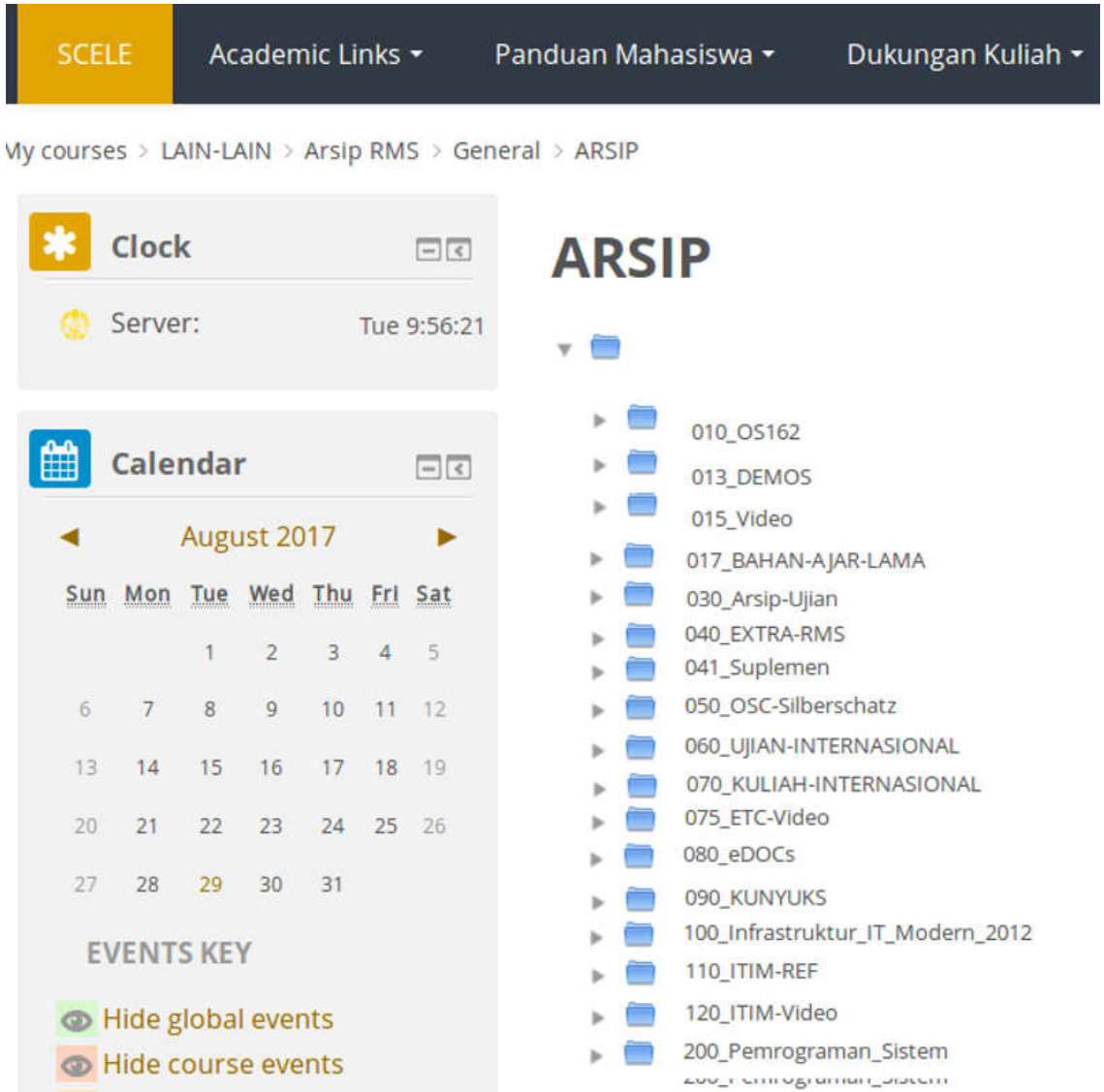


Figure: Lihat juga BADAK.cs.ui.ac.id:/extra/

Bahan presentasi: http://rms46.vlsm.org/2/207.html

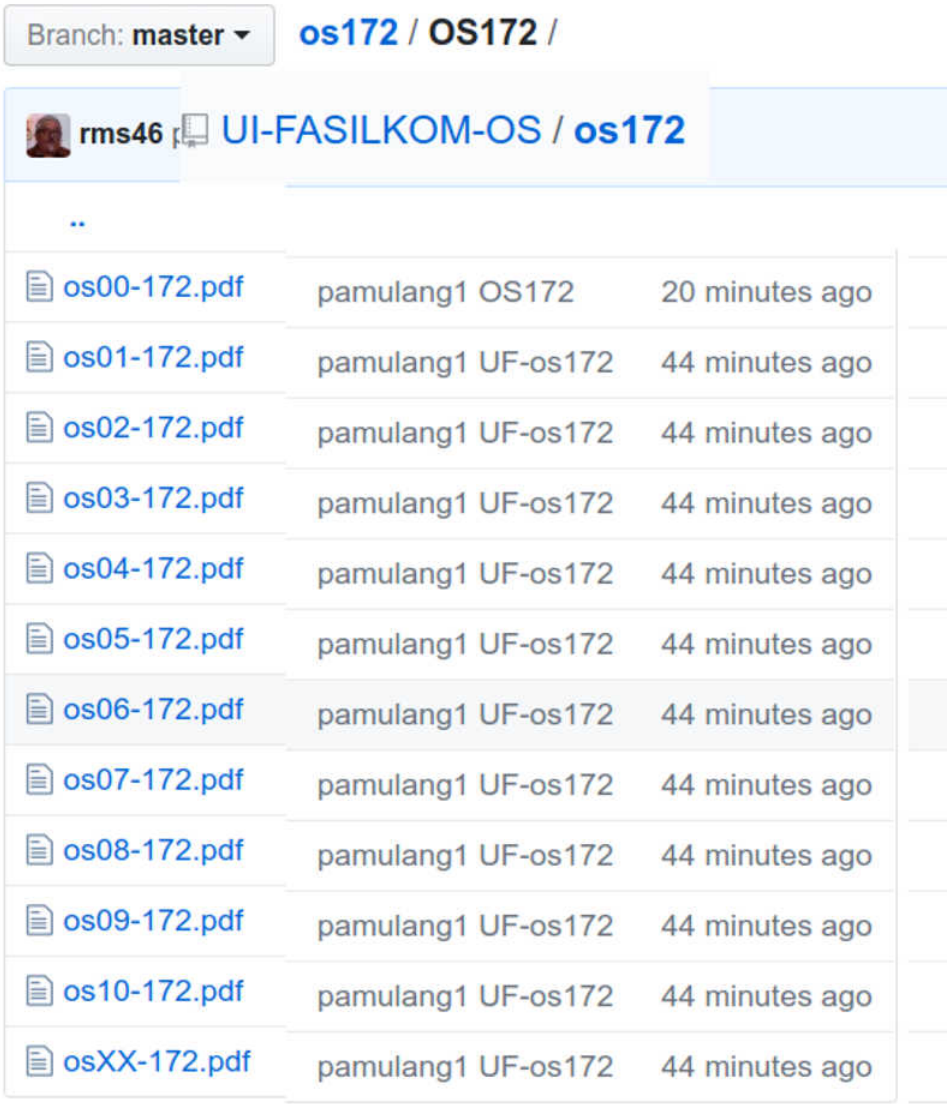


Figure: Alternatif: BADAK.cs.ui.ac.id:/extra/OS172/

BADAK.cs.ui.ac.id:/extra/

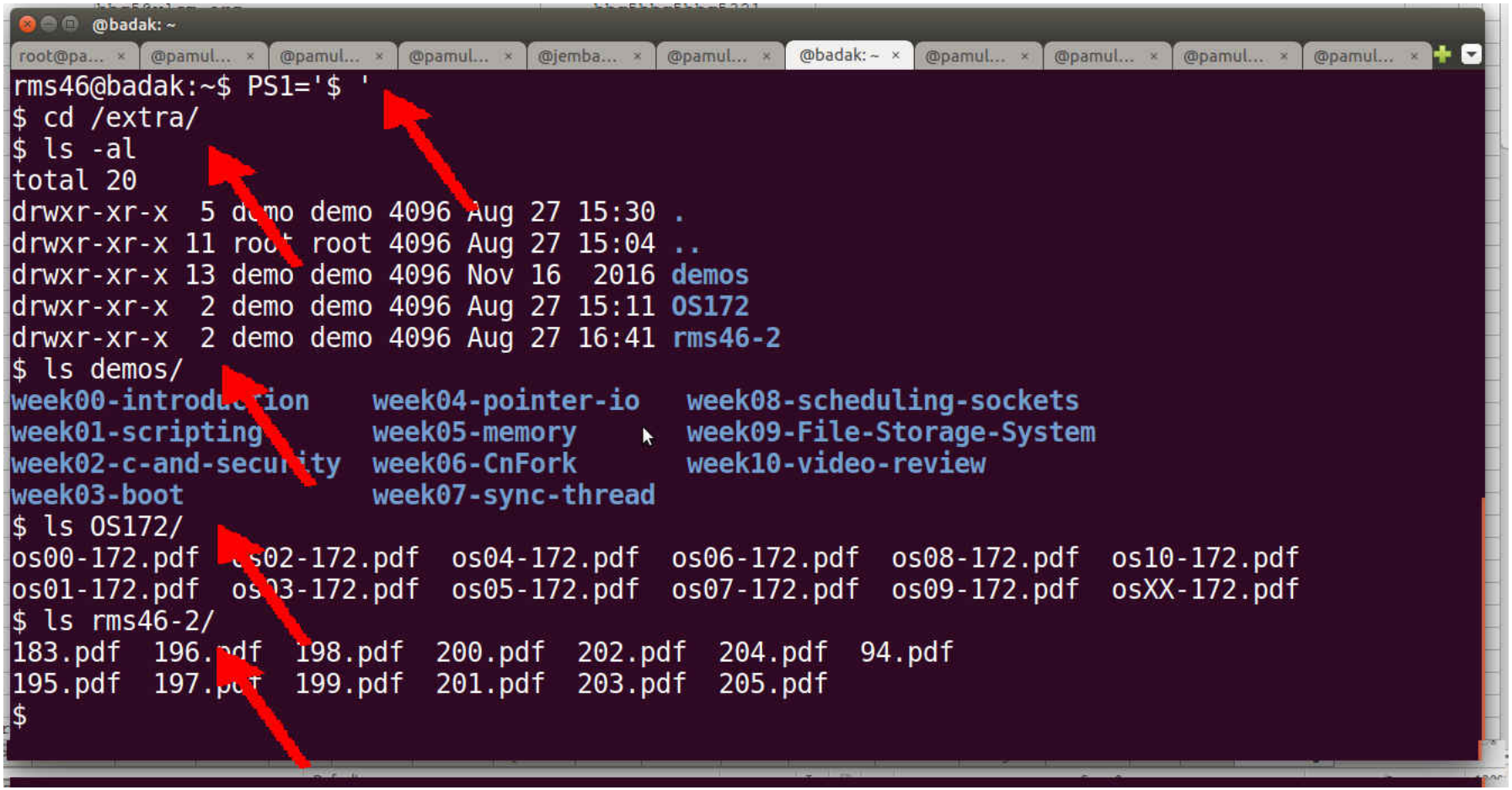


Figure: BADAK.cs.ui.ac.id:/extra/

Self Service Lab

- **Informasi Pendaftaran**
  - Pengarahan Pendaftaran: 2 September 2017 jam 10:00
    `https://scele.cs.ui.ac.id/mod/forum/discuss.php?d=5313`
  - Tutorial: `http://bit.ly/Week00`
- Informasi Kuliah, Arsip Ujian, dan Demo
  - `badak.cs.ui.ac.id:/extra/`
  - `https://github.com/UI-FASILKOM-OS/os172`
  - `https://github.com/UI-FASILKOM-OS/demo`
  - `https://rms46.vlsm.org/2/195.pdf` — [195.pdf - 205.pdf].
- BASH Account:
  - Virtual Ubuntu: badak.cs.ui.ac.id (SSO)
  - Ubuntu (BYOD)
  - WSL: Windows 10 Subsystem for Linux
  - Cygwin (Windows)
- Memo Mingguan / QR Code.

Memo Mingguan

- Demo, Lab dan Scele: **TIDAK WAJIB**[1]
- **WAJIB**: mempersiapkan/mempelajari bahan kuliah minggu terkait.
- **WAJIB**: mempersiapkan selembar kertas memo dengan *QR Code*.
  - Format QRC: "OS172 KELAS NPM NAMA_SIAK".
  - Kelas: A (reguler), E (Extention), I (International), M (Matriculation), X (ETC).
  - Contoh QRC: "OS172 X 9999999999 Cicak bin Kadal".
  - Ukuran QRC: sekitar 400x400 pixel.
  - Memo berisi catatan secukupnya (lihat contoh).
- Harap **TEST** apakah QRC terbaca (dengan aplikasi QRC reader).
- Memo akan di-*scan* di kelas pada kuliah pertama minggu terkait[2].
- Blanko Memo sama dapat digunakan pada minggu berikutnya.

[1] namun sangat direkomendasikan untuk kelas E I M!

[2] kecuali kuliah minggu 00.

Contoh Memo Minggu 00

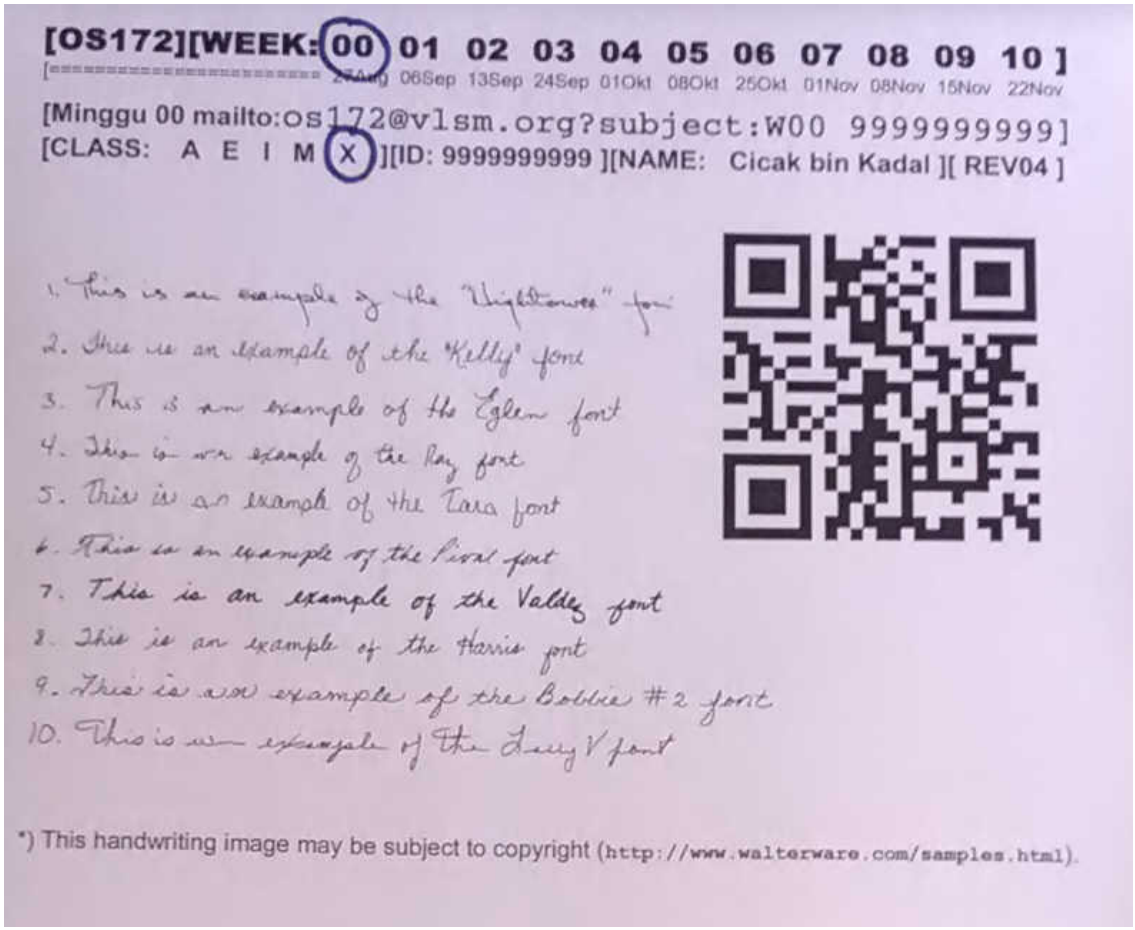


Figure: **Kertas Ukuran A4**: OS172 X 9999999999 Cicak bin Kadal

Pembuatan dan Pembacaan QR Code

```
# OS172: OS 2017 2nd term
# CLASS: A (reguler), E (Extention), I (International),
#        M (Matriculation), X (ETC).
# ID:    Student ID (NPM)
# NAME:  NAME (SIAK)

$ qrencode "OS172 X 9999999999 Cicak bin Kadal" -s 20 -o
    OS172-A-9999999999.png

$ zbarimg OS172-A-9999999999.png
QR-Code:OS172 X 9999999999 Cicak bin Kadal
scanned 1 barcode symbols from 1 images in 0.11 seconds
```

Quick Response (QR) Code

Administratip/ETC

- QRCode Memo:
  - tujuan pengiriman memo ialah sebagai "bukti" telah belajar.
  - besar harapan pada saat datang ke kelas:
    - telah mempersiapkan diri dengan membuat memo secukupnya.
    - telah memahami garis besar bahan minggu terkait.
    - telah mempelajari jenis soal UTS/UAS yang pernah ditanyakan pada masa lalu.
  - ISI memo tidak dinilai!
  - Memo yang baik ialah MEMO yang bermanfaat untuk pembuatnya.
  - Ujian:
    - Saat UTS dipersilakan membawa hingga 6 lembar memo ber QRC.
    - Saat UAS dipersilakan membawa hingga 5 lembar memo ber QRC.
    - Memo boleh yang pernah ditulis atau boleh juga membuat memo (ber QRC) yang baru.
- Harap menghubungi SEKRE (Ged. B lt. 2) untuk segala masalah administratip, terutama absen, sakit, surat sakit, ujian susulan, dst.
- Harap merampungkan masalah administrasi ujian susulan dalam 6 hari kerja.

WSL 1

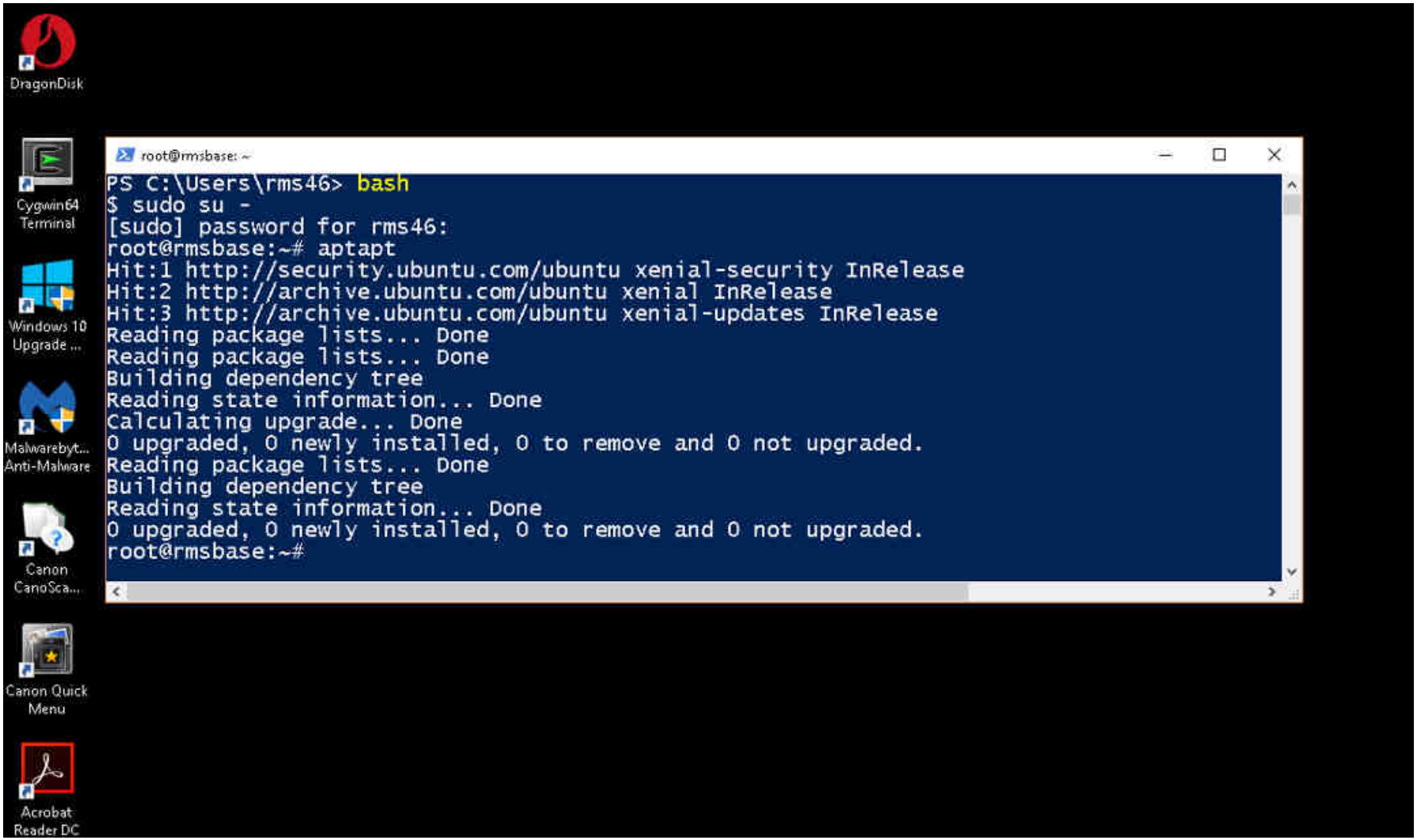


Figure: WSL: Windows Subsystem for Linux

WSL 2

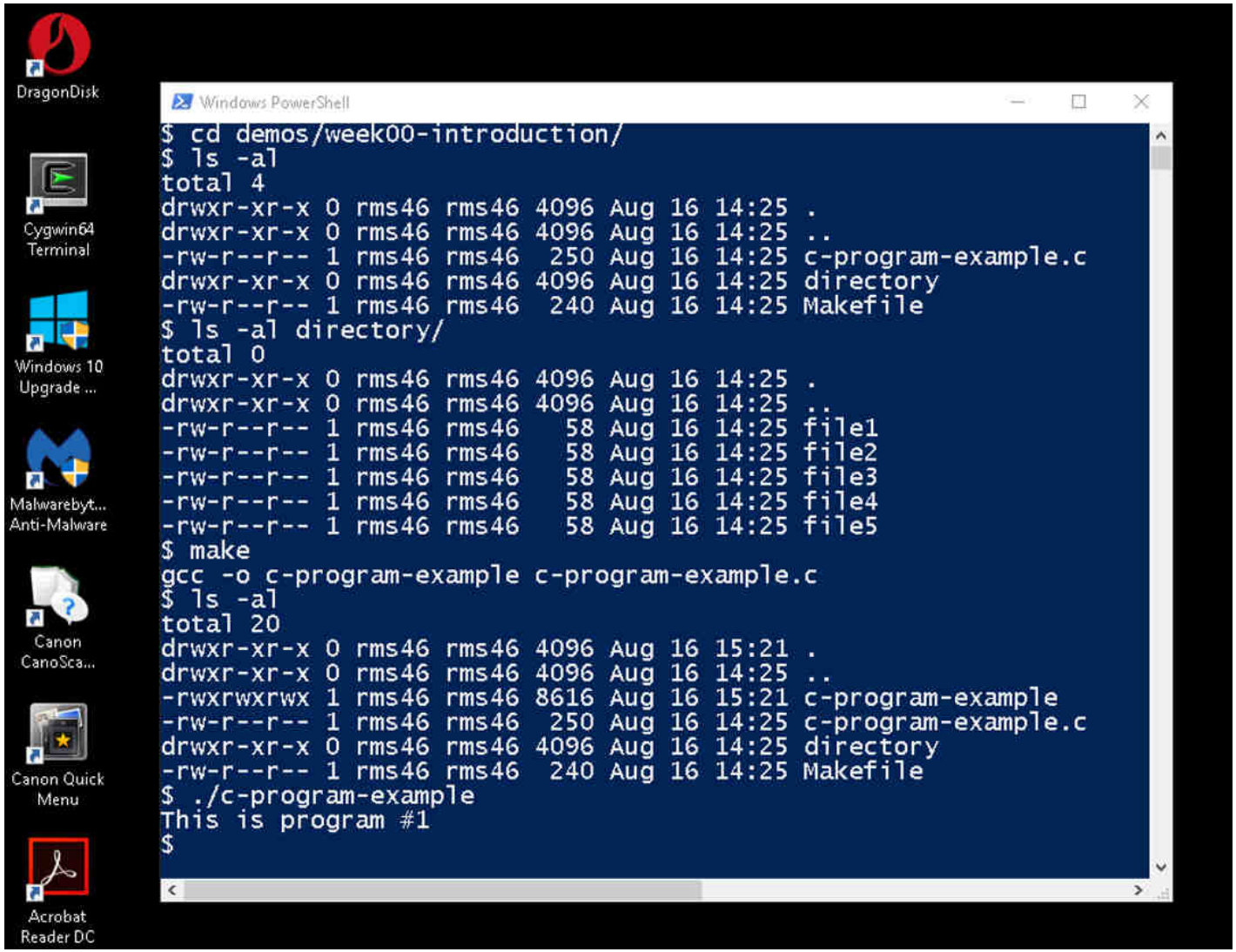


Figure: WSL: Windows Subsystem for Linux

Cygwin

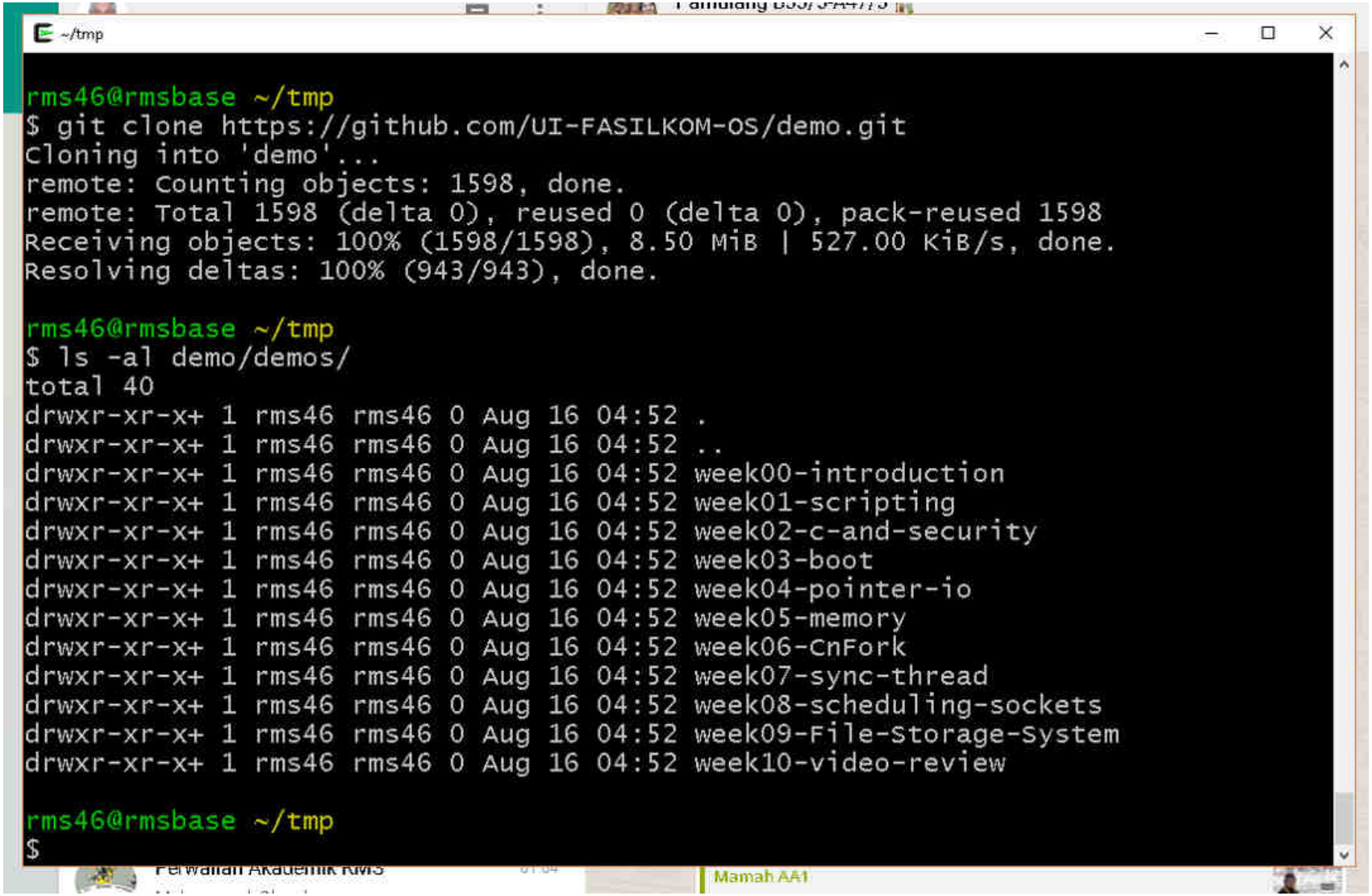


Figure: Cygwin

Cygwin

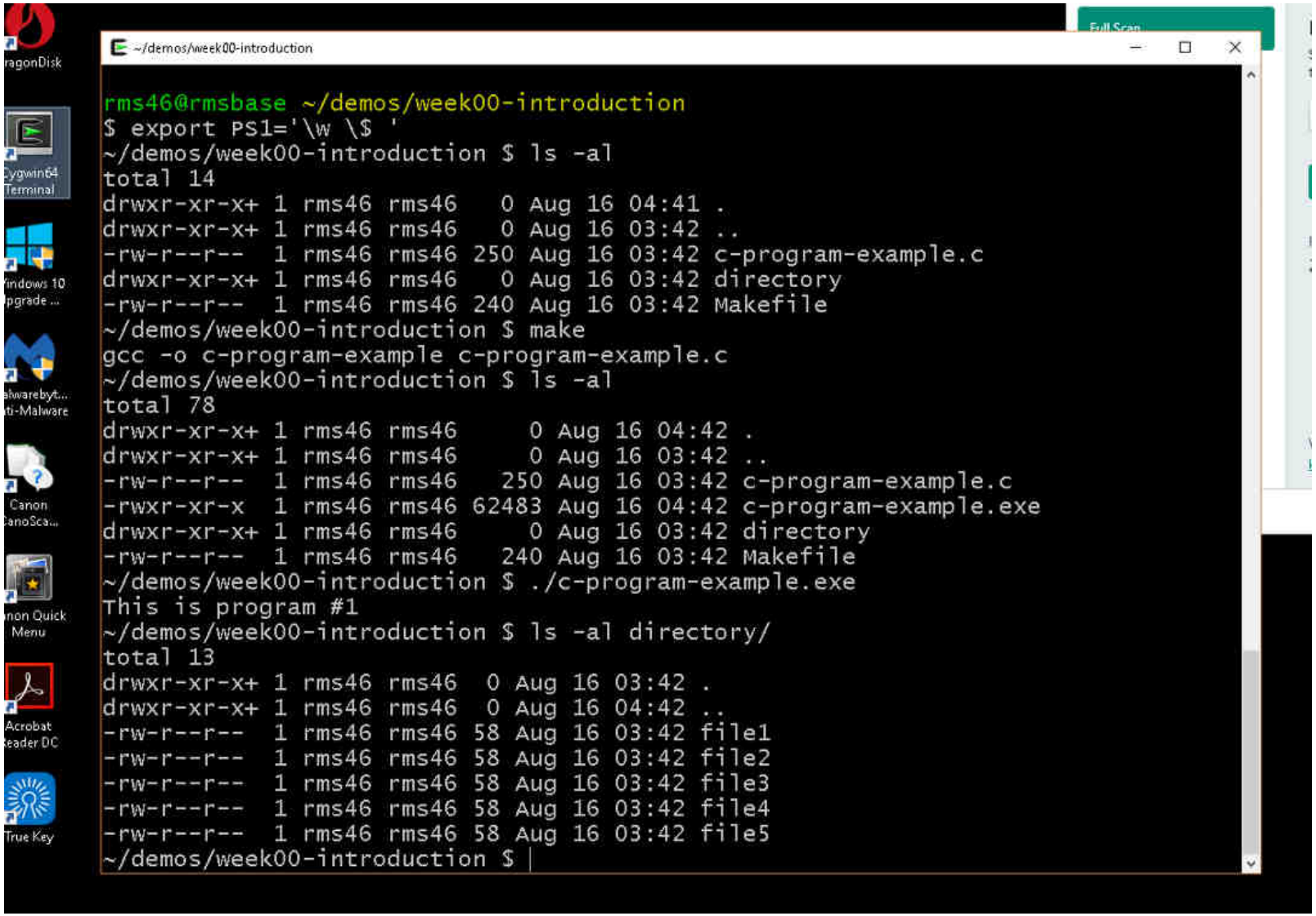


Figure: Cygwin

Github Demo

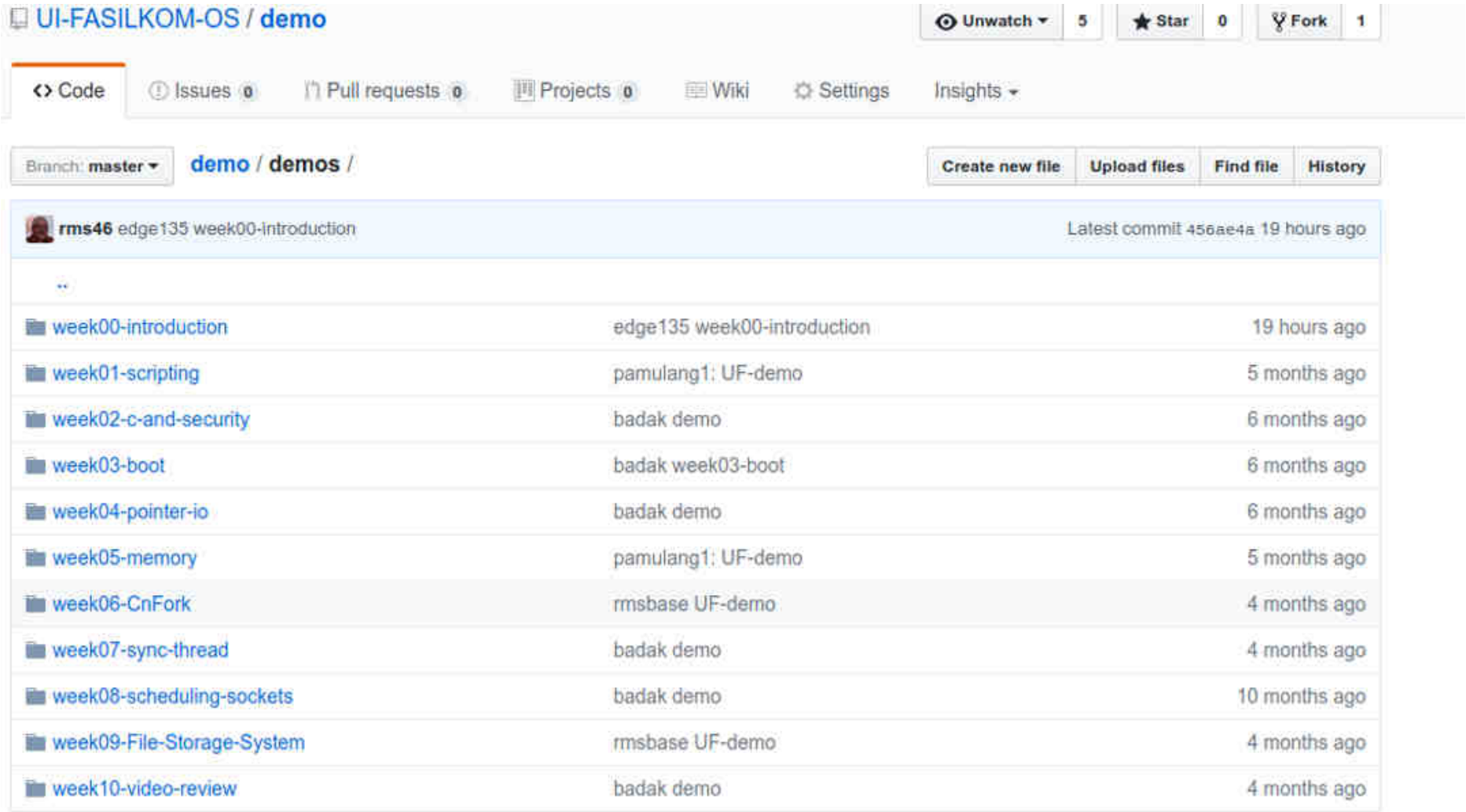


Figure: `https://github.com/UI-FASILKOM-OS/demo`

Login: Kawung dan Badak

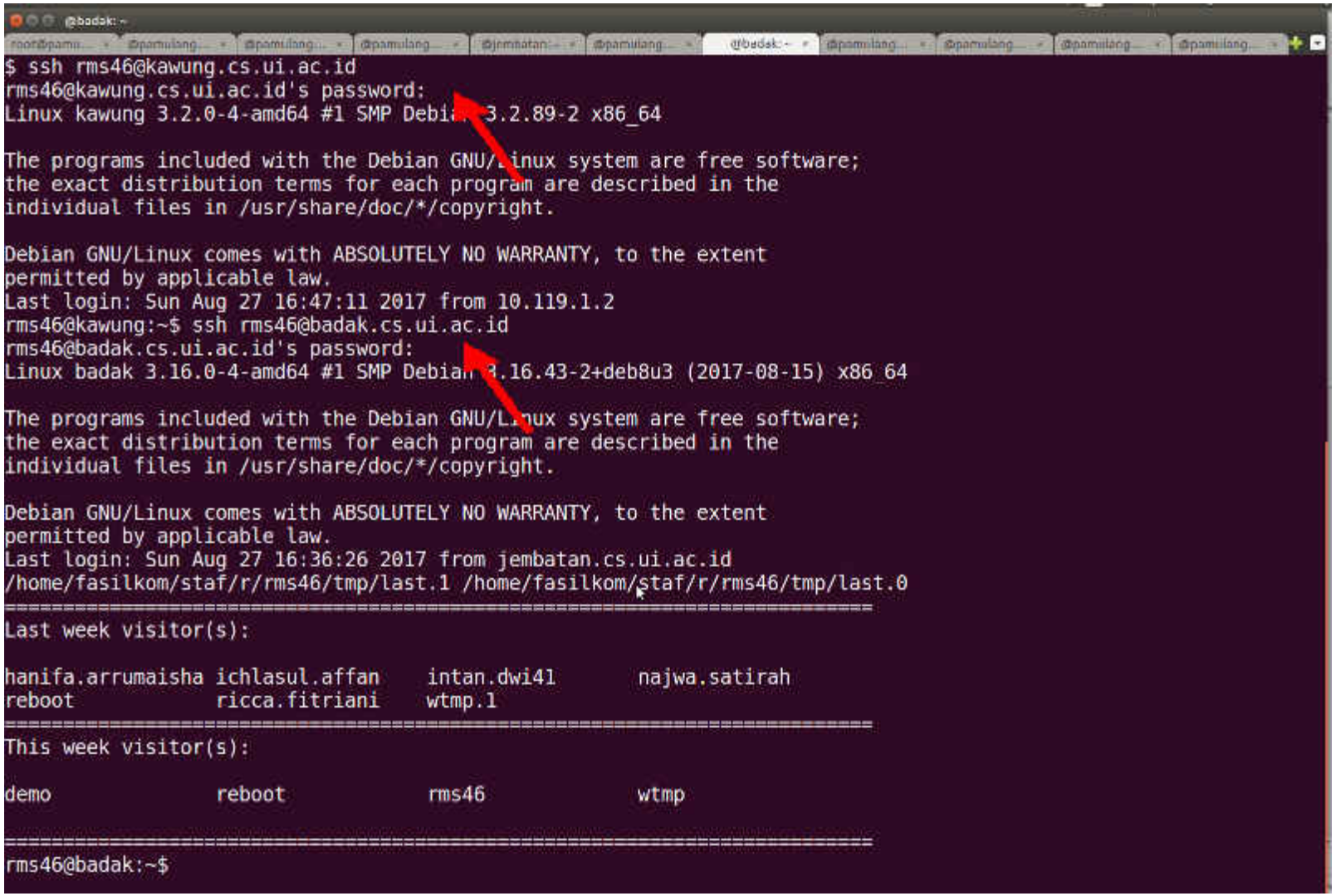


Figure: Login: Kawung dan Badak

Program Example

```
$ cat c-program-example.c
/* (c) 2016-2017 Rahmat M. Samik-Ibrhaim
 *  REV01 Sun Aug 20 15:01:12 WIB 2017
 *  START Fri Jan 01 00:00:00 WIB 2016
 *  This is a free software.
 *  To compile:
 *  $ gcc -o c-program-example c-program-example.c
 *  To execute:
 *  $ ./c-program-example
 */

#include <stdio.h>

void main() {
   printf("This is program #1\n");
}
```

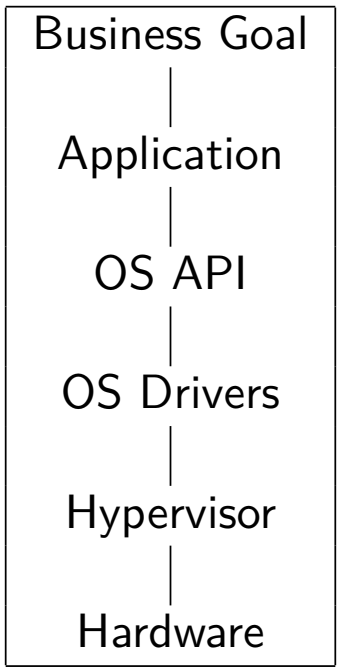
Makefile

```
$ cat Makefile

# (c) 2016-2017 Rahmat M. Samik-Ibrahim
# REV01 Tue Aug 22 14:45:14 WIB 2017
# START Fri Jan 01 00:00:00 WIB 2016
# This is a free Makefile configuration.
# Just run:
# % make

ALL:  c-program-example

c-program-example: c-program-example.c
   gcc -o c-program-example c-program-example.c

clean:
   rm -f c-program-example
```

```
$ ls -al
total 44
drwxr-xr-x  3 rms46 rms46  4096 Aug 28 18:45 .
drwxr-xr-x 13 rms46 rms46  4096 Feb 28 18:50 ..
-rw-r--r--  1 rms46 rms46   334 Aug 23 20:17 c-program-example.c
drwxr-xr-x  2 rms46 rms46  4096 Sep 26  2016 directory
-rw-r--r--  1 rms46 rms46   319 Aug 23 20:17 Makefile
-rw-r--r--  1 rms46 rms46 23606 Aug 28 18:26 QuickResponseCode
                                                        .docx
$ make
gcc -o c-program-example c-program-example.c
$ ./c-program-example
This is program #1
$ ls -al
total 56
........
$ make clean
rm -f c-program-example
$
```

Week 00: Introduction

- Reference: (OSCE2e ch1/2)(UCB 01)(UDA P1L1/2)(OLD 00)
- Operating System
  - Why take this OS class?
  - Definition: Resource Allocator & Control Program.
  - Managers: Process, Memory, Storage, ...
  - Layers
  - Interfaces



Computer Organization Review

- You should understand:
  - von Neumann Model.
  - Buses, Bridges, Transfer Rate, Clock.
  - Memory: DDR, DDR-2, ...
  - Cache, Buffer, Spool.
  - Direct Memory Access (DMA).
  - Port & Memory Mapped I/O.
  - CPU: privilege/kernel/supervisor mode and user mode.
  - Hardware Limitation.
  - Priority: Read vs Write.
  - Interrupts: Polling & Vectored.
  - Multiprocessors: Symmetric vs. Asymmetric.
  - Multicore & Multithreading.
  - Clustered Systems.
  - Numbers: base 2, base 8, base 10, base 16.
    - Base 2: 110010101010_2
    - Base 8: $01234567_8 = 000\ 001\ 010\ 011\ 100\ 101\ 110\ 111_2$
    - Base 10: 012 345 679
    - Base 16: $9AB\ CDEF_{16} = 1001\ 1010\ 1011\ \ 1100\ 1101\ 1110\ 1111_2$

Block Diagram

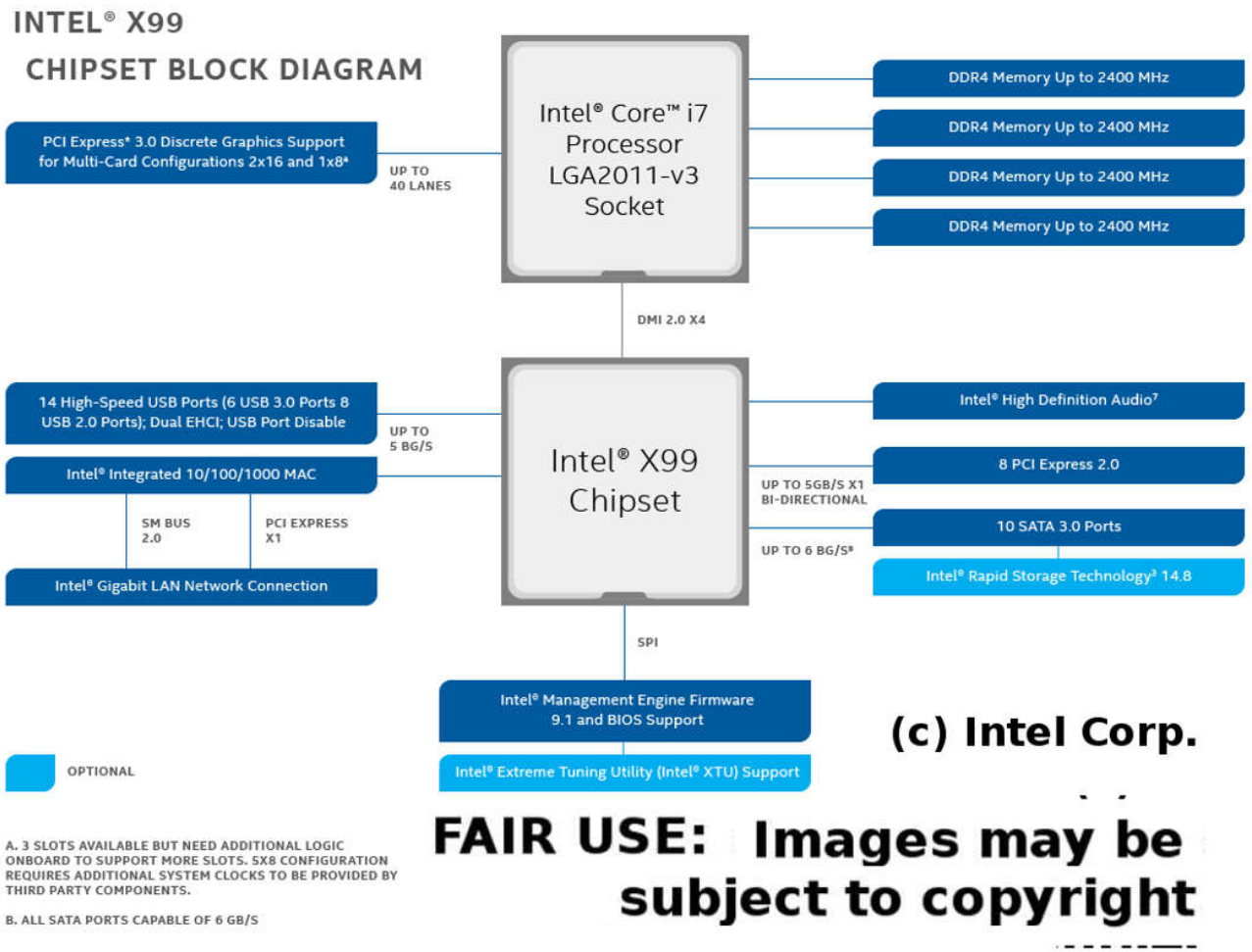


Figure: Block Diagram

APIC

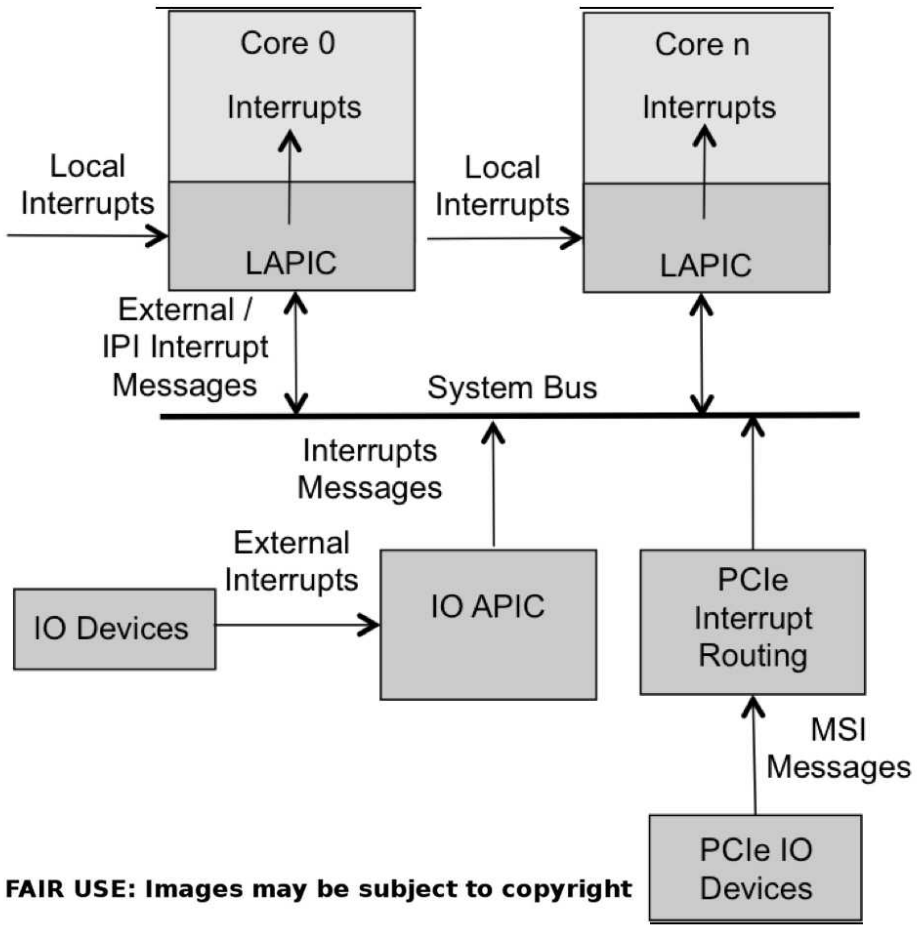


Figure: APIC (Advanced Programmable Interrupt Controller)

Interupt Handling

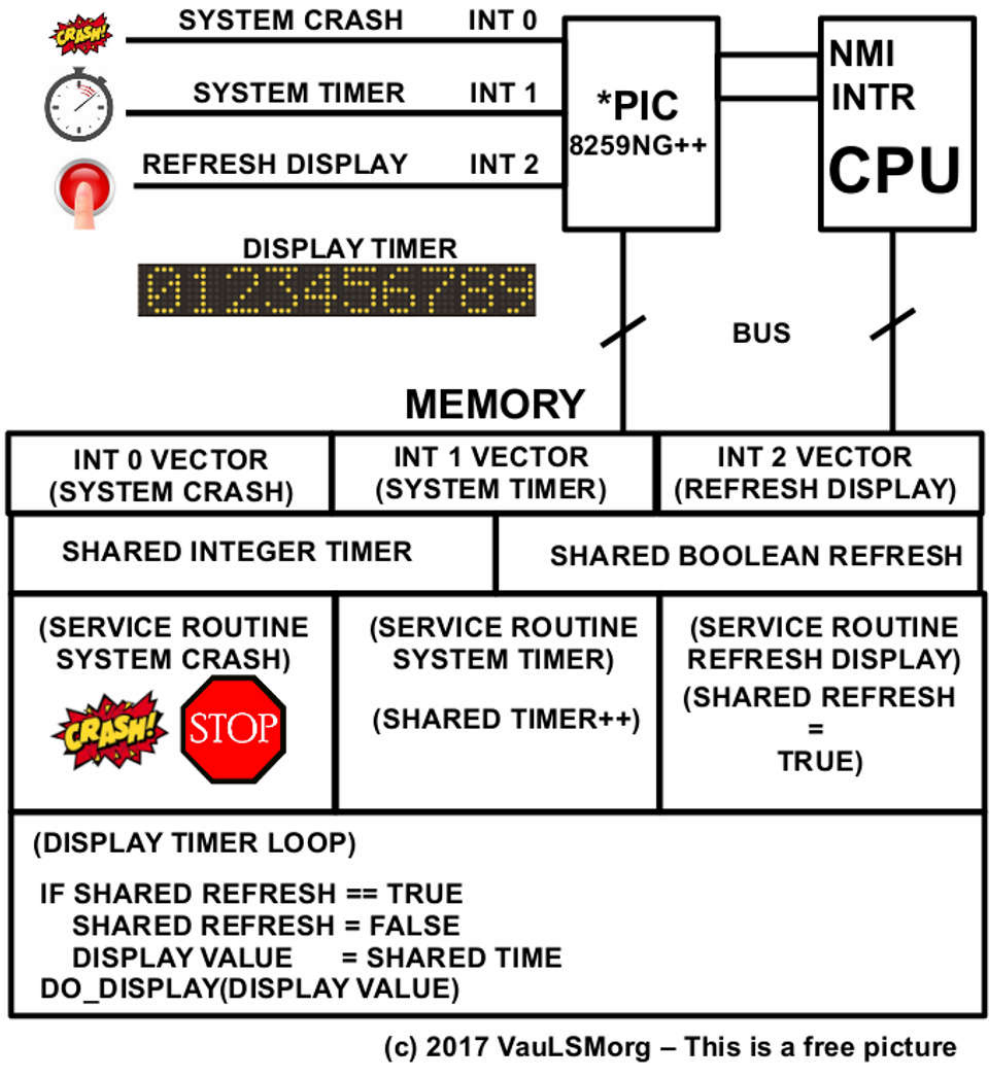


Figure: Interupt Handling with PIC (Programmable Interrupt Controller)

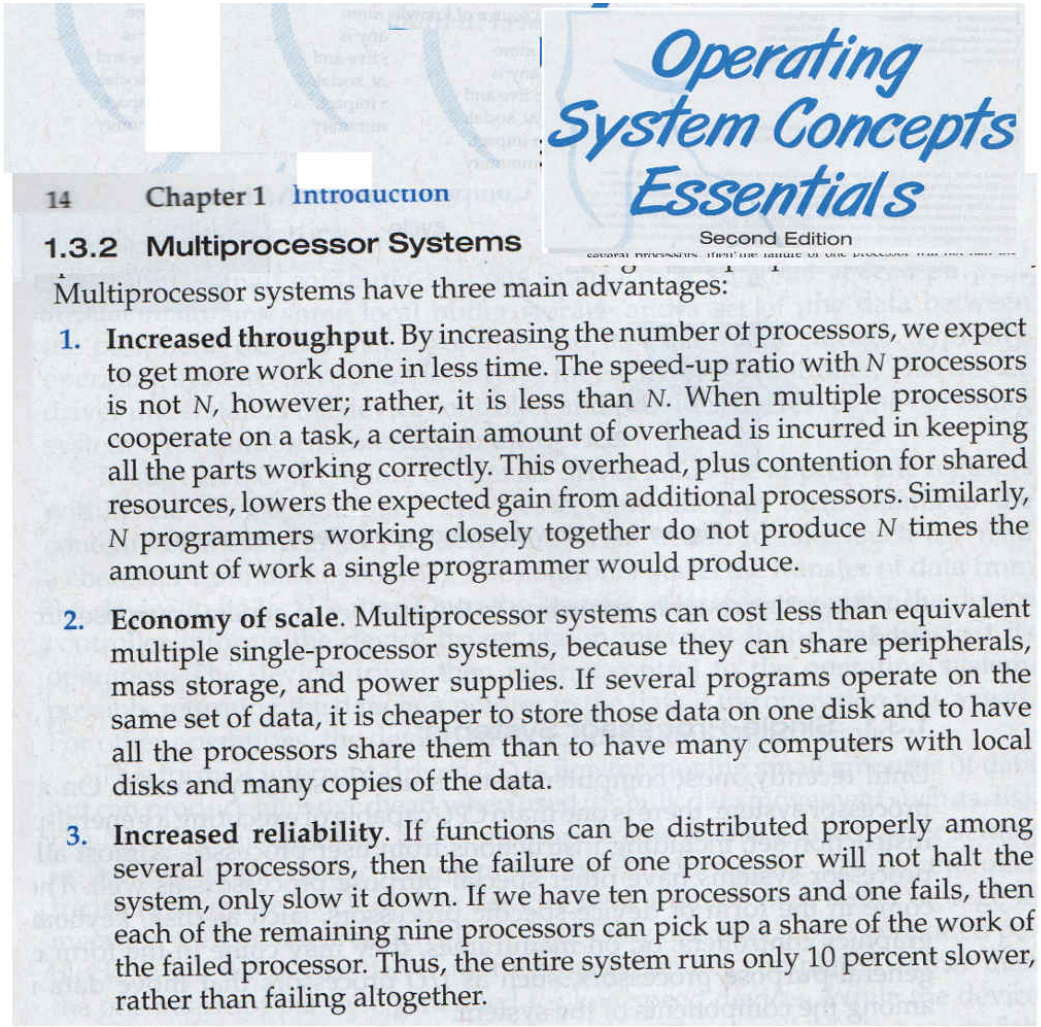
Managers Set

- Process:
  - Creating/Deleting; Suspending/Resuming; Synchronization; Communication; Schedulling
- Memory:
  - Tracking; Move In/Move Out; Allocating/Deallocating.
- Storage/File System:
  - Create/Delete; Open/Close; Read/Write.
- Mass Storage:
  - Schedulling; Allocating; Free Space.
- I/O:
  - Buffering; Caching; Spooling.
  - Interfacing (driving).
- Protecting & Security:
  - Protecting.
  - Security.

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.

Aneka Soal Ujian Sistem Operasi Week00 2016-1 (OSC2e)



Operating System Concepts Essentials
Second Edition

14 Chapter 1 Introduction

1.3.2 Multiprocessor Systems

Multiprocessor systems have three main advantages:

1. **Increased throughput**. By increasing the number of processors, we expect to get more work done in less time. The speed-up ratio with N processors is not N, however; rather, it is less than N. When multiple processors cooperate on a task, a certain amount of overhead is incurred in keeping all the parts working correctly. This overhead, plus contention for shared resources, lowers the expected gain from additional processors. Similarly, N programmers working closely together do not produce N times the amount of work a single programmer would produce.
2. **Economy of scale**. Multiprocessor systems can cost less than equivalent multiple single-processor systems, because they can share peripherals, mass storage, and power supplies. If several programs operate on the same set of data, it is cheaper to store those data on one disk and to have all the processors share them than to have many computers with local disks and many copies of the data.
3. **Increased reliability**. If functions can be distributed properly among several processors, then the failure of one processor will not halt the system, only slow it down. If we have ten processors and one fails, then each of the remaining nine processors can pick up a share of the work of the failed processor. Thus, the entire system runs only 10 percent slower, rather than failing altogether.

Figure: **T / F** The advantages of a multiprocessor system include: increased throughput, economy of scale, and increased reliability.

The End

- This is the end of the presentation.