CSF2600505 Sistem Operasi
CSGE602055 Operating Systems
Week 00: Overview 1

Rahmat M. Samik-Ibrahim

University of Indonesia

`http://rms46.vlsm.org/2/207.html`
`Always check for the latest revision!`

REV131 11-Apr-2018

Operating Systems 2018-1 (Room 3114 Tue/Thu)
Class: A (10:00-12:00) | B (13:00-15:00) | C (16:00-18:00)

Week	Schedule	Topic	OSC9
Week 00	06 Feb - 12 Feb 2018	Overview 1	Ch. 1, 16
Week 01	13 Feb - 19 Feb 2018	Overview 2 & Scripting	Ch. 1, 2
Week 02	20 Feb - 26 Feb 2018	Protection, Security, Privacy, & C-language	Ch. 14, 15
Week 03	27 Feb - 05 Mar 2018	I/O, BIOS, Loader, & Systemd	Ch. 13
Week 04	06 Mar - 12 Mar 2018	Addressing, Shared Lib, & Pointer	Ch. 8
Week 05	13 Mar - 19 Mar 2018	Virtual Memory	Ch. 9
Reserved	20 Mar - 24 Mar 2018		
Mid-Term	03 Apr 2018	13:00 - 15:30 (UTS)	
Week 06	05 Apr - 11 Apr 2018	Concurency: Processes & Threads	Ch. 3, 4
Week 07	12 Apr - 18 Apr 2018	Synchronization	Ch. 5, 7
Week 08	19 Apr - 25 Apr 2018	Scheduling	Ch. 6
Week 09	26 Apr - 07 May 2018	File System & Persistent Storage	Ch. 10, 11, 12
Reserved	08 May - 14 May 2018		
Week 10	15 May - 21 May 2018	I/O Programming & Network Sockets Programming	
Reserved	22 May - 22 May 2018		
Final	23 May - 26 May 2018	(UAS)	
Deadline	07 Jun 2018 16:00	Extra assignment **deadline**	

- The Check List (Operating Systems)
  - **Starting Point**: `http://rms46.vlsm.org/2/207.html`
  - **Text Book**: any recent/decent OS book but map it to **OSC9**.
  - Create **public** project "os181" on your `github.com` account.
    - Create file "`README.md`" and add an extra line every week. For e.g.[1]:

      ```
      ZCZC Sistem Operasi 2018 Awal (1)
      ZCZC W01 Have tried demo for week 01.
      ZCZC W02 Week 02 is done.
      ZCZC W03 Week 03 is done.
      ```

  - Encode your **QRC** with image size of approximately 250x250 pixels: **"OS181 CLASS ID GITHUB-ACCOUNT SSO-ACCOUNT SIAK-Full-Name"** Special for Week 00: Mail your **embedded** QRC to: `os181@vlsm.org` with Subject: [W00] CLASS ID SIAK-NAME.
  - Write your Memo (with QRC) **every week**.
  - Using your **SSO** account, login to `badak.cs.ui.ac.id` via `kawung.cs.ui.ac.id`.
    - Check folder `badak:///extra/Week00/`
    - Every week, copy the weekly demo files to your own home directory. Eg. for Week00:
      `cp -r /extra/Week00/W00-demos/ W00-demos/`

[1] Week 00 line is optional. The following "ZCZC WXX" weekly tags are mandatory.

Agenda

1. Start
2. Agenda
3. How to contact the Lecturer
4. Goal
5. Assessment
6. Schedule
7. Resources
8. Week 00: Self Service Assignments
9. Encoding and Decoding a QR Code
10. Memo Mingguan + QRC
11. Bahan-bahan
12. Accounts
13. Week 00: Review
14. Week 00: Problems
15. Week 00: Summary
16. Week 00: Check List
17. The End

How to contact the Lecturer[2]

- Kontak/Tanya/Jawab WhatsApp Group **OS181**
  (info +62-881-456-**XXXX**)
  Email (Subject:[HELP]) os181@vlsm.org

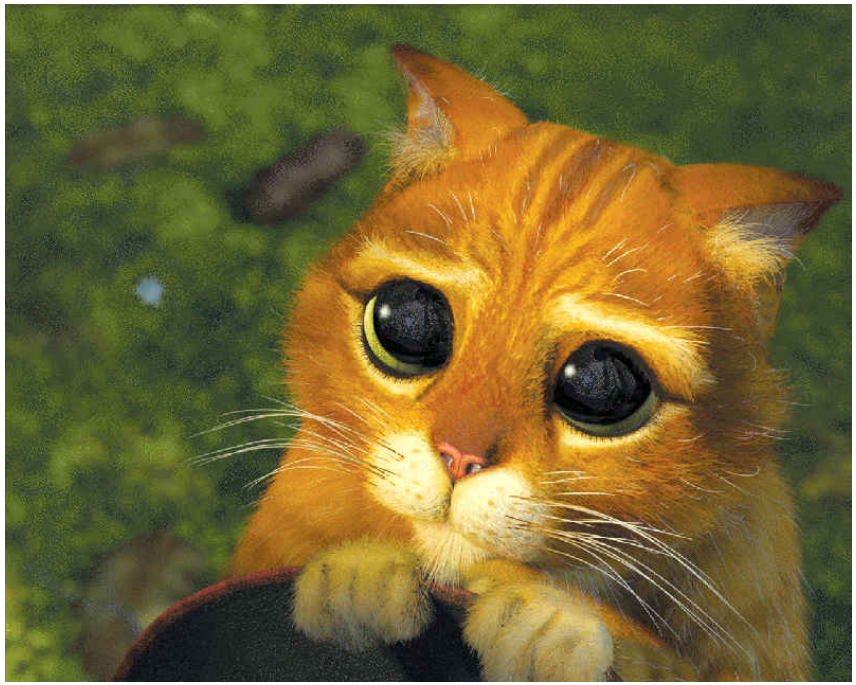


Figure: This is Puss in Boot[1].

[1] This is a fair use of a DreamWorks/Paramount Picture character.
[2] FYI: King Goerge II founded the University of Goettingen in 1734.

Goal

Coverage

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

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 Pop Quiz – No Teaching Assistant**.
- **Active Preparation / Participation / Q&A Only**.
  - Pre-Midterm (UTS): 6 weeks @ 3 points (=18%).
  - Post-Midterm: 5 weeks @ 3 points (=15%).
  - Points for answering questions, trying demos, and writings memos.
  - Deductions for **NOT** answering questions: individually or collectively.
- UTS: 6 set problems @ 6 points (=36%).
- UAS: 5 set problems @ 6 points (=30%).
- Extra untuk nilai C keatas: 1 point[1].
- C-2C untuk nilai C-: upto 5 points[1].

[1] Syarat dan Ketentuan Berlaku

Schedule pre MidTerm (UTS)

- Week00 – Overview 1 (OSC9-ch01[1] OSC9-ch16).
- Week01 – Overview 2 & Scripting (OSC9-ch01 OSC9-ch02 Scripting PLB-70 demo-w01[2]).
- Week02 – Protection, Security, Privacy, & C-language (OSC9-ch14 OSC9-ch15 demo-w02).
- Week03 – I/O, BIOS, Loader, & Systemd (OSC9-ch13 demo-w03).
- Week04 – Addressing, Shared Lib, & Pointer (OSC9-ch08 demo-w04).
- Week05 – Virtual Memory (OSC9-ch09 demo-w05).
- MidTerm (UTS) Week 00 — 05.

[1] OSC9: Operating Systems Concepts (9^{th} edition).
[2] Demo Files.

Schedule post MidTerm (UTS)

- Week06 – Concurency: Processes & Threads (OSC9-ch03 OSC9-ch04 demo-w06).
- Week07 – Synchronization (OSC9-ch05 OSC9-ch07 demo-w07).
- Week08 – Scheduling (OSC9-ch06 demo-w08).
- Week09 – File System & Persistent Storage (OSC9-10 OSC9-ch11-OSC9-ch12 demo-w09).
- Week10 – I/O Programming & Network Sockets Programming (demo-w10).
- Final (UAS) Week 06 — 10.

Resources

- Buku Sistem Operasi yang terbit dalam 10 tahun terakhir, umpama: (OSC9) Abraham Silberschatz, Peter B. Galvin, Greg Gagne: Operating System Concepts, 9^{th} Edition, 2013.
- SUP — (ARSIP)(041_Suplemen) Supplement.
- ETC — (ARSIP)(075_ETC-Video) ETC
- (GITHUB) **https://github.com/UI-FASILKOM-OS/os181**
  - (DEMO) — demos/
  - (SLIDE) — pdf/ — **http://rms46.vlsm.org/2/207.html**
- (UJIAN) — **http://rms46.vlsm.org/2/195.pdf - 205.pdf**
- ARCHIVE (Arsip bahan pengajaran): **https://scele.cs.ui.ac.id/course/view.php?id=126**
  - **Enrollment key:** "11100100"[1].
- (BADAK) — BADAK:///extra/

[1] Kunci akan berubah secara berkala.

Week 00: Self Service Assignments

- What is your class? A? B? C? D? E? I? M? X?
- Create project (**PUBLIC**) "os181" on your new (or existing) github.com account.
- Check your existing SSO Account (for using badak.cs.ui.ac.id).
- (Week 00) QRCode[1]: "OS181 CLASS ID GITHUB-ACCOUNT SSO-ACCOUNT SIAK-Full-Name"
- (Weekly) Memo.
- Informasi Kuliah, Arsip Ujian, dan Demo
  - `badak.cs.ui.ac.id:/extra/`
  - `https://github.com/UI-FASILKOM-OS/os181`
  - `https://rms46.vlsm.org/2/195.pdf` — [195.pdf - 205.pdf].
- Which BASH Account?
  - Virtual Ubuntu: badak.cs.ui.ac.id (SSO)
  - Ubuntu (BYOD)
  - WSL: Windows 10 Subsystem for Linux
  - Cygwin (Windows)

[1] "QR Code" is a registered trademark and wordmark of Denso Wave Inc.

Encoding and Decoding a QR Code

```
# OS181:           OS 2018 1st term
# CLASS:           A, B, C, D (reg.), E (Extension), I (International),
#                  M (Matriculation), X (ETC). Eg. "M".
# ID:              Student ID (NPM). Eg. "1253759225"
# GITHUB-ACCOUNT: Student's GITHUB acount. Eg. "cbkadal"
# SSO-ACCOUNT:    Student's SSO acount. Eg. "cicak123"
# SIAK-FULL-NAME: Student's SIAK name. Eg. "Cicak Bin Kadal"

$ qrencode "OS181 M 1253759225 cbkadal cicak123 Cicak Bin Kadal" \
    -s 7 -o OS181-M-1253759225.png

$ zbarimg OS181-M-1253759225.png
QR-Code:OS181 M 1253759225 cbkadal cicak123 Cicak Bin Kadal
scanned 1 barcode symbols from 1 images in 0.11 seconds
```

Quick Response Code (QRC) Code
Suggested size: 256 x 256 pixel.
Check yours with a QRC reader app.

Administratip

- (Week 00 **only**): Send QRC to `os181@vlsm.org`
  - Subject: [W00] CLASS ID SIAK-NAME
  - Example:
    - Mailto: os181@vlsm.org
    - Subject: [W00] M 1253759225 Cicak Bin Kadal
    - Insert your QR Code (**embedded**).
- Masalah Administratip
  - 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.

Memo Mingguan + QRC

- **WAJIB**: mempersiapkan/mempelajari bahan kuliah minggu **setiap** terkait.
  - telah memahami garis besar bahan minggu terkait.
  - telah mempelajari jenis soal UTS/UAS yang pernah ditanyakan pada masa lalu.
- Telah mempersiapkan diri dengan membuat memo yang ada *QRC*.
  - Harap **TEST** apakah QRC terbaca dengan aplikasi QRC reader.
  - QRC memo akan di-*scan* di kelas pada saat istirahat kuliah pertama minggu terkait[1].
  - tujuan pembuatan memo ialah sebagai "bukti" telah belajar.
  - 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.

[1] kecuali kuliah minggu 00.

Week 00 Memo Example

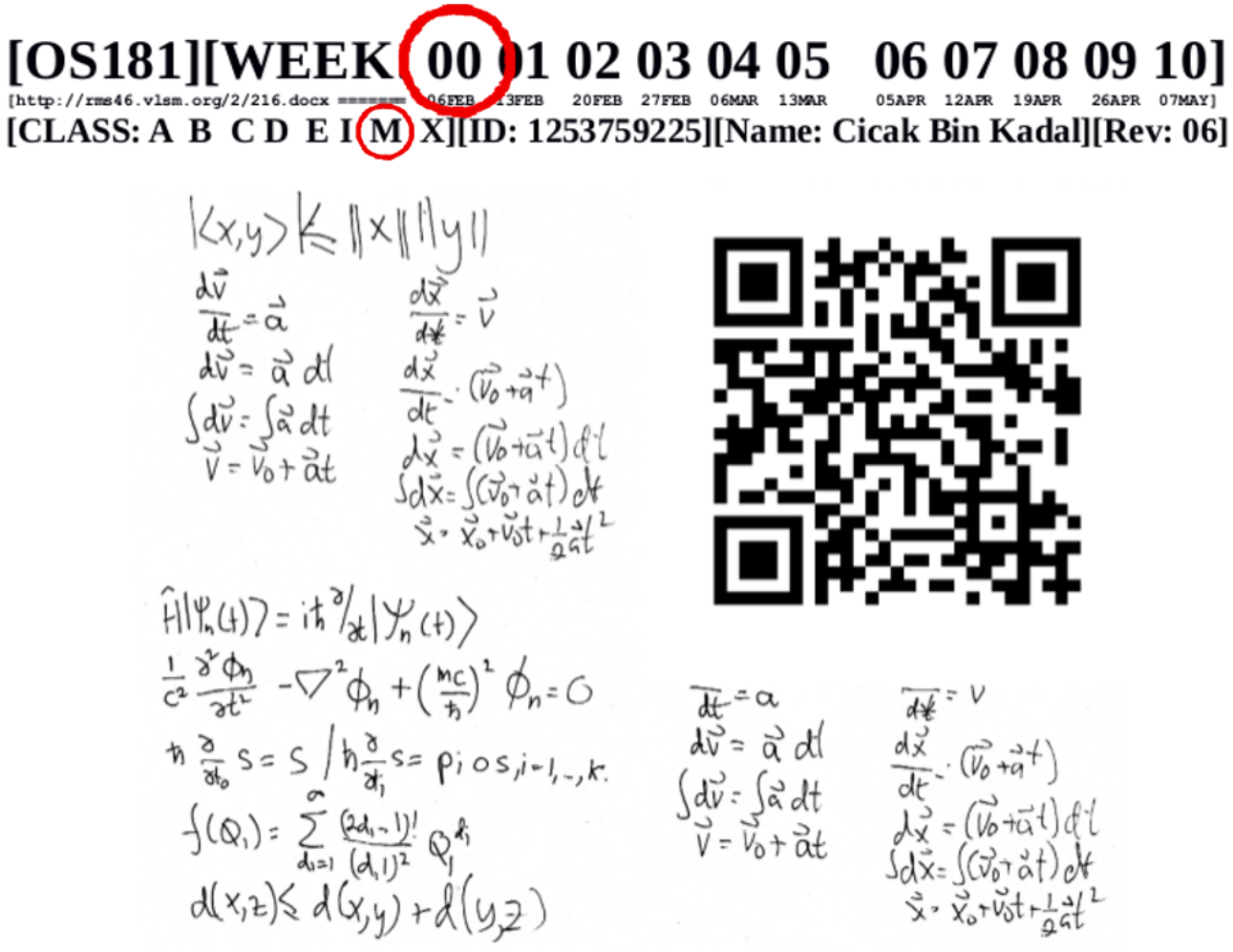


Figure: **Memo**: OS181 M 1253759225 cbkadal cicak123 Cicak Bin Kadal

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

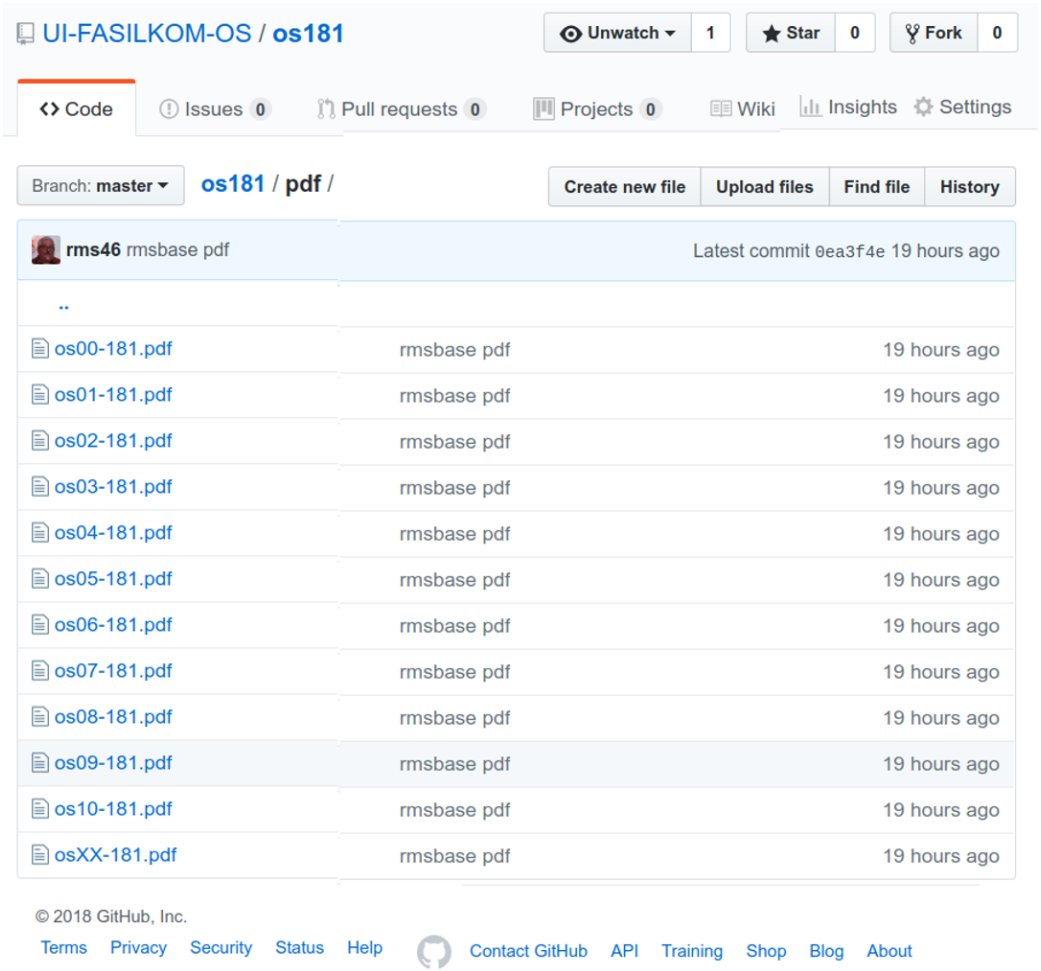


Figure: https://github.com/UI-FASILKOM-OS/os181/tree/master/pdf

Bahan Demo

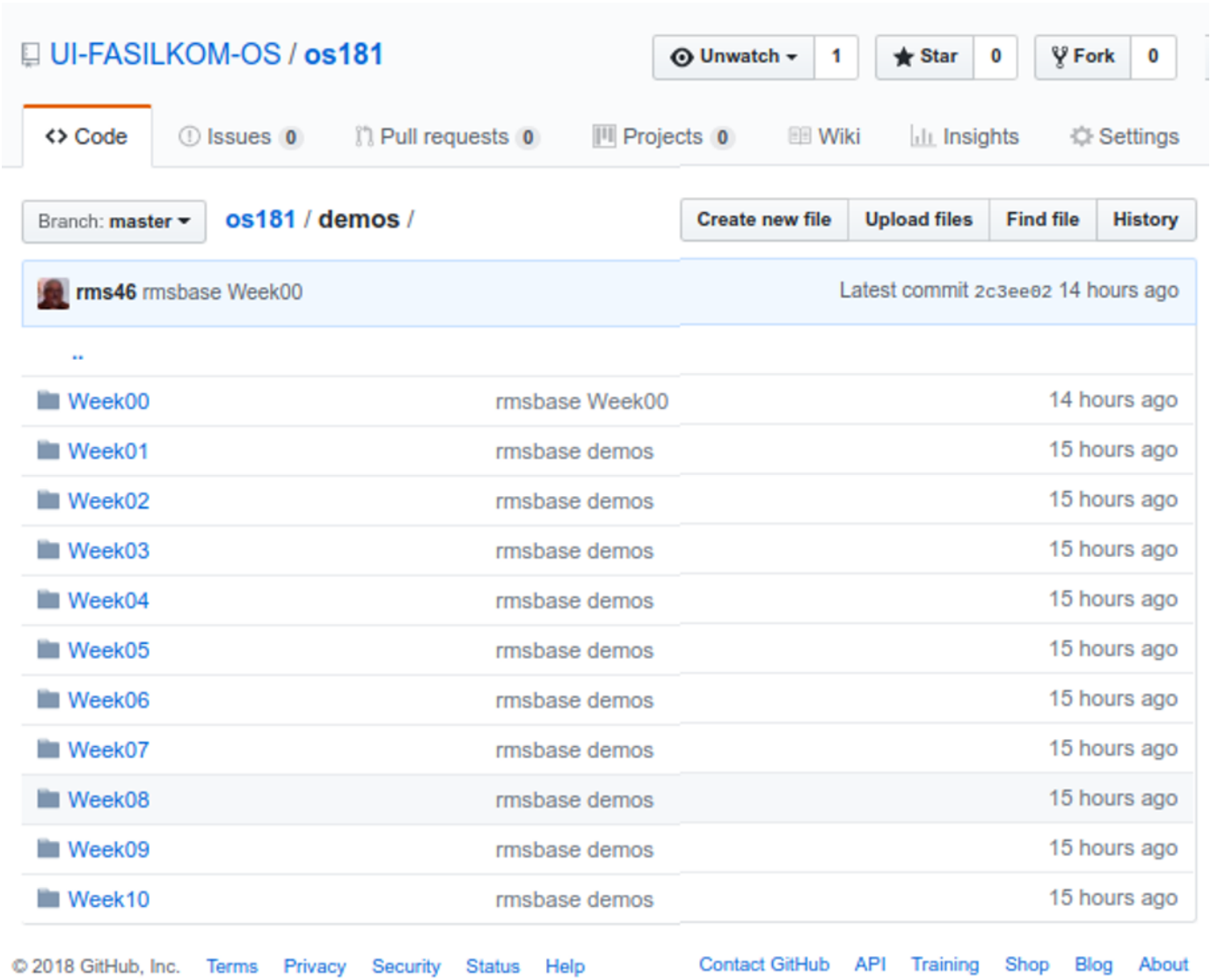


Figure: `https://github.com/UI-FASILKOM-OS/os181/tree/master/demos`

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

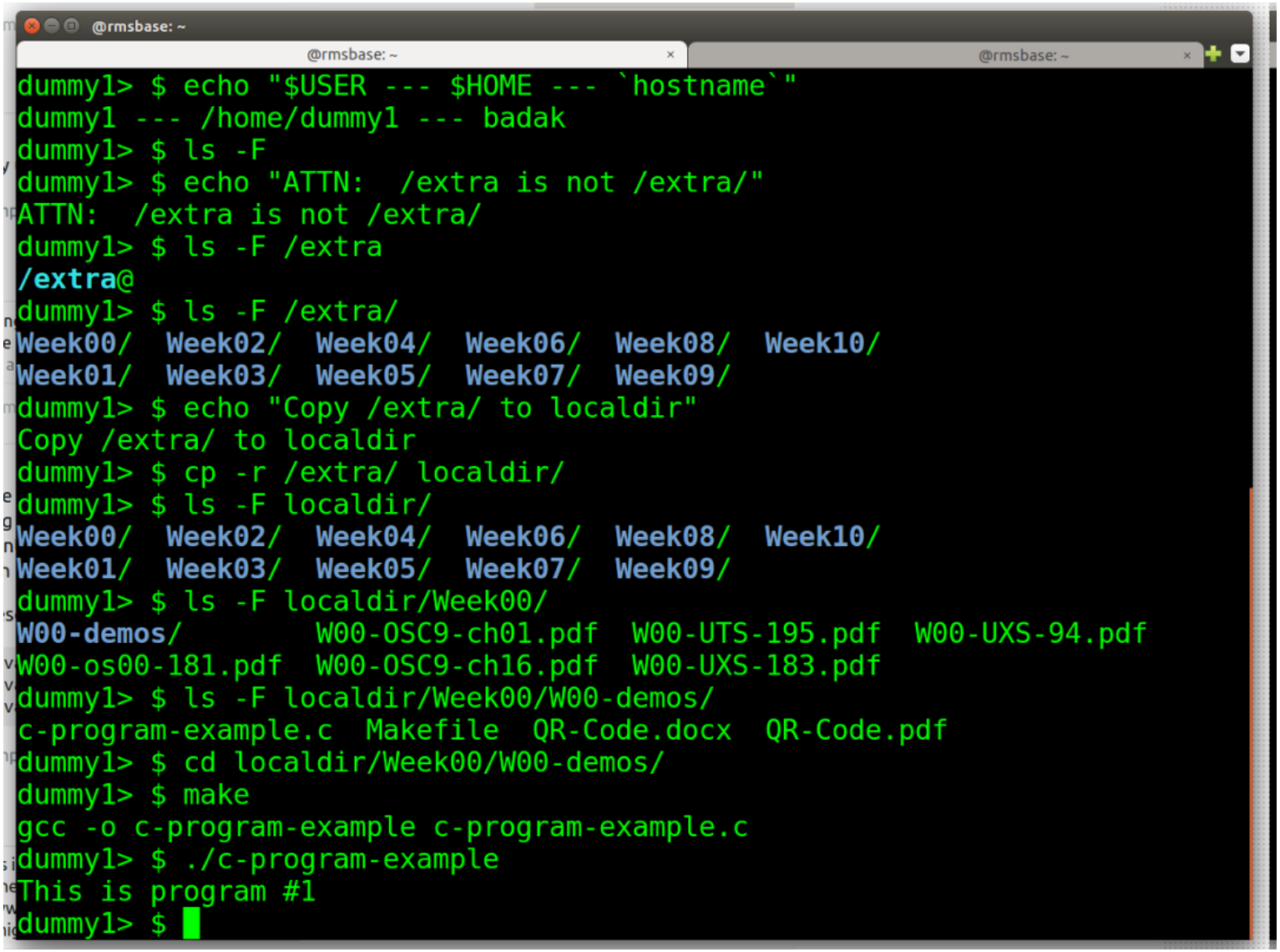


Figure: BADAK.cs.ui.ac.id:///extra/

Arsip SCELE

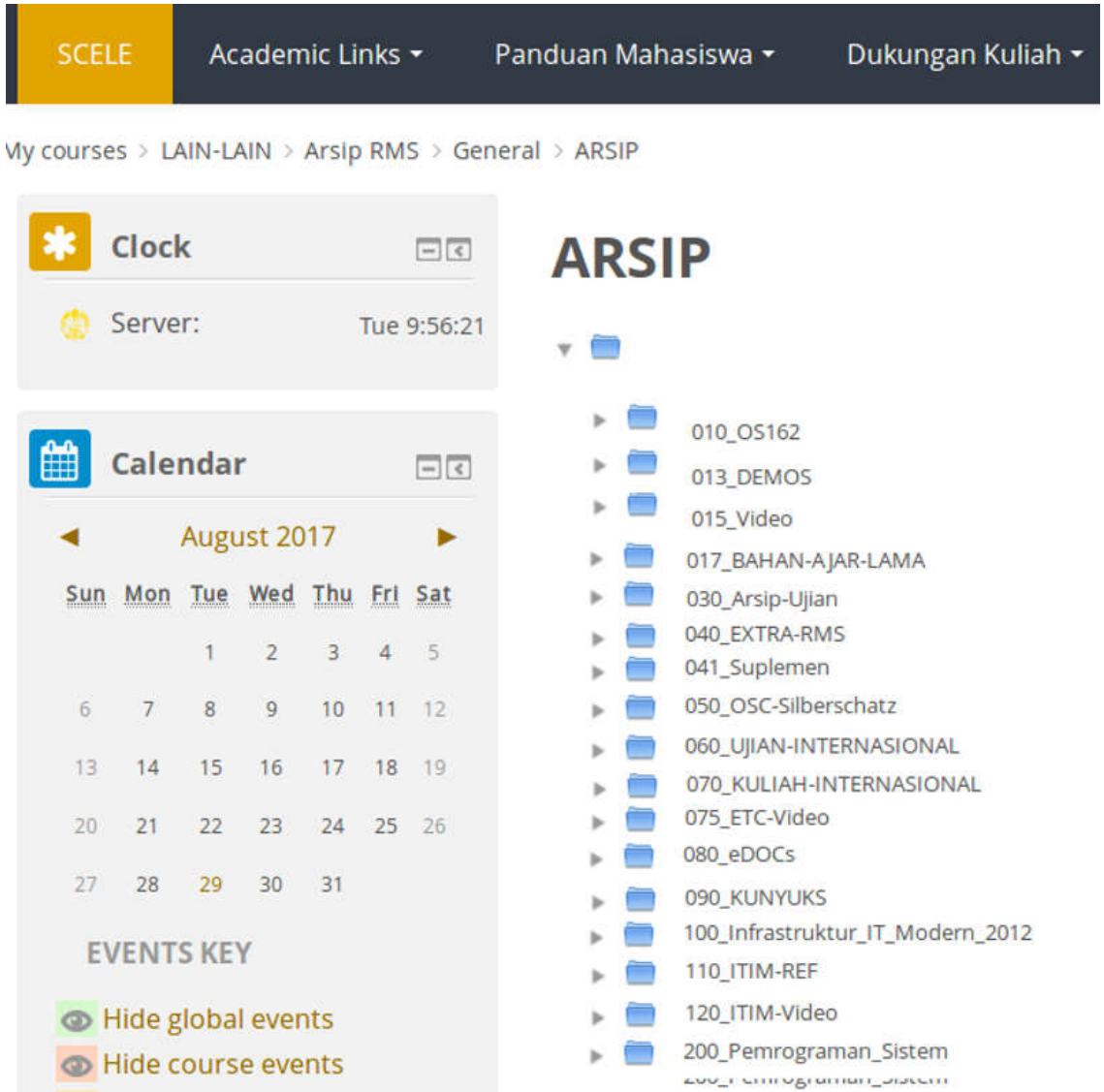


Figure: Lihat juga BADAK.cs.ui.ac.id:///extra/

Github (New) Account 1

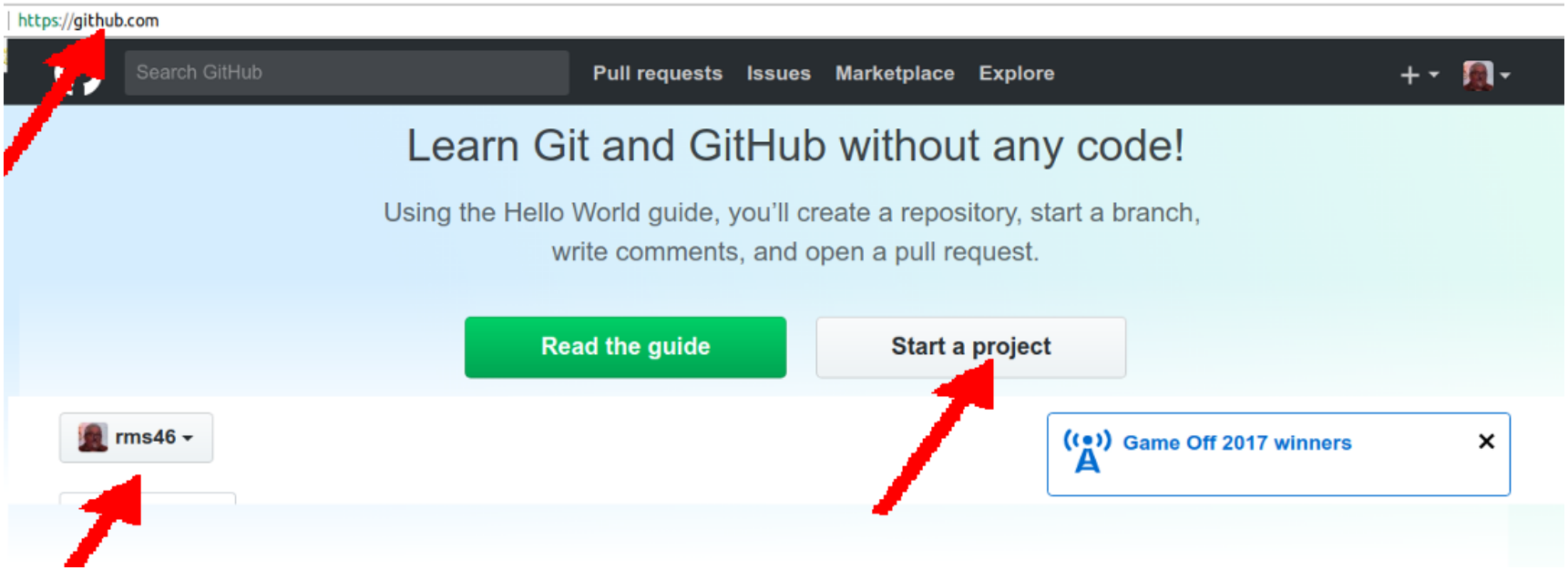


Figure: Start a new project by "rms46".

Github (New) Account 2

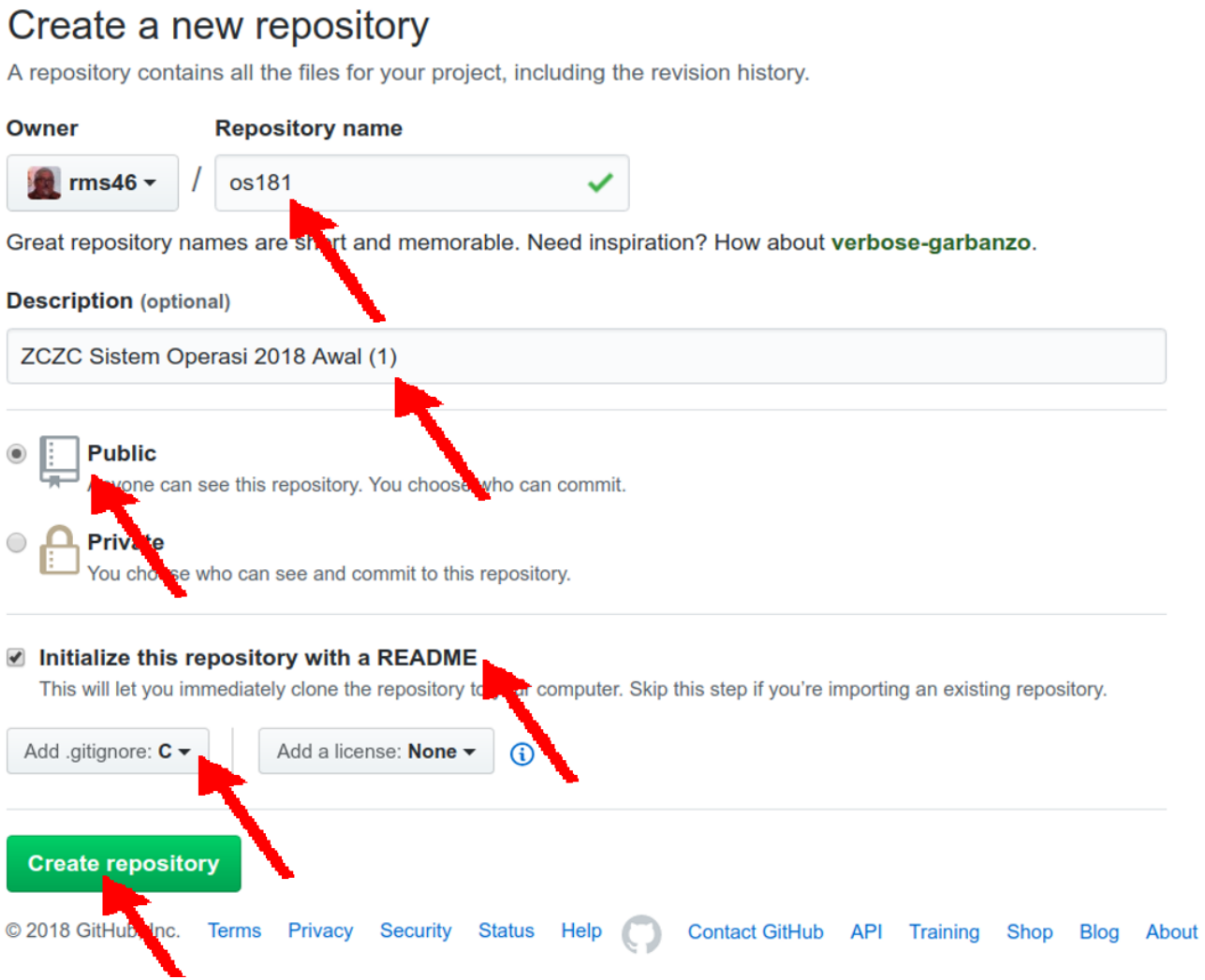


Figure: Create public repository "os181" with a README.md file

Github (New) Account 3

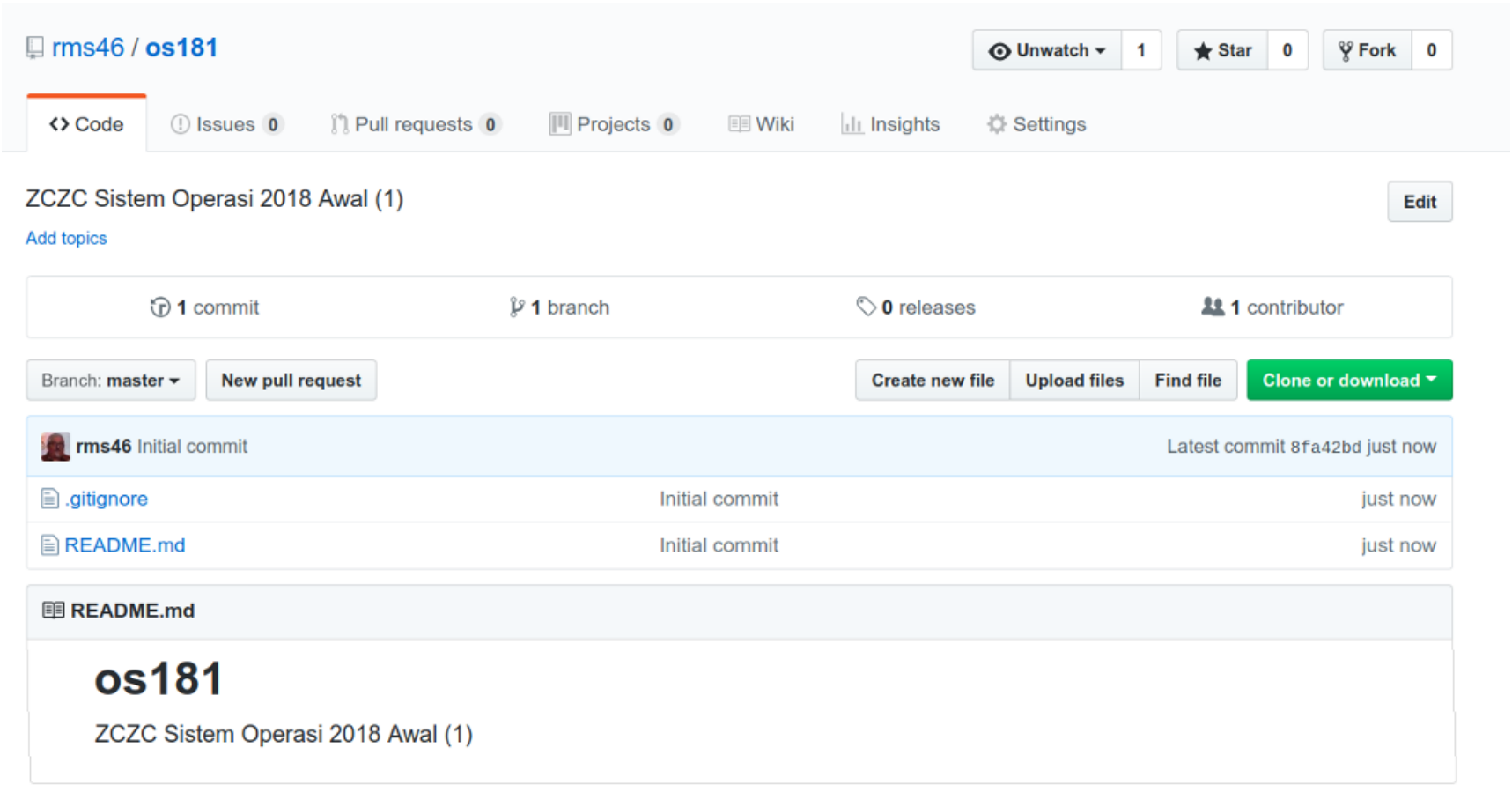


Figure: Public project "os181" by "rms46" at https://github.com/rms46/os181

Login: Badak via Kawung

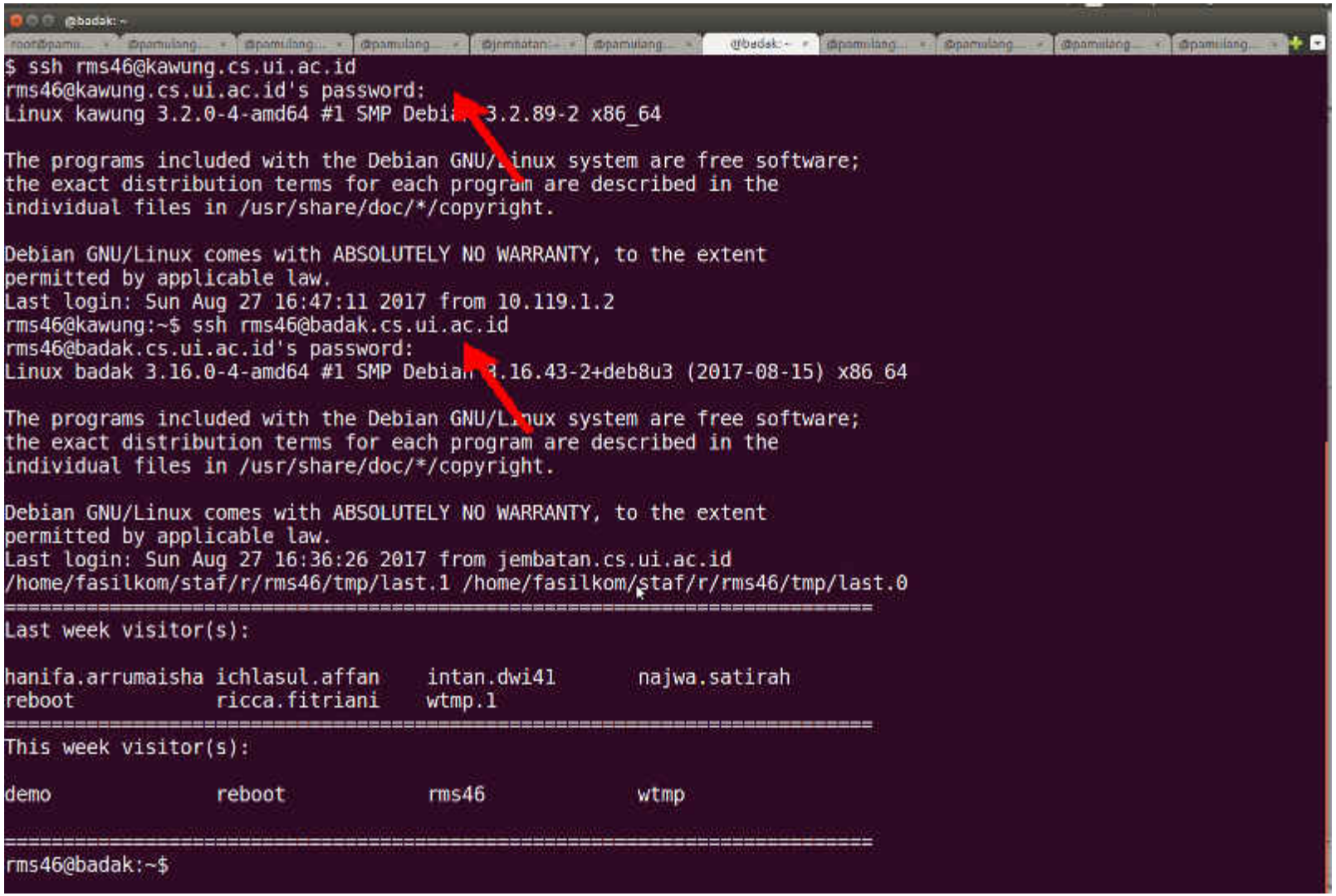


Figure: Login: Badak via Kawung

WSL: Windows Subsystem for Linux

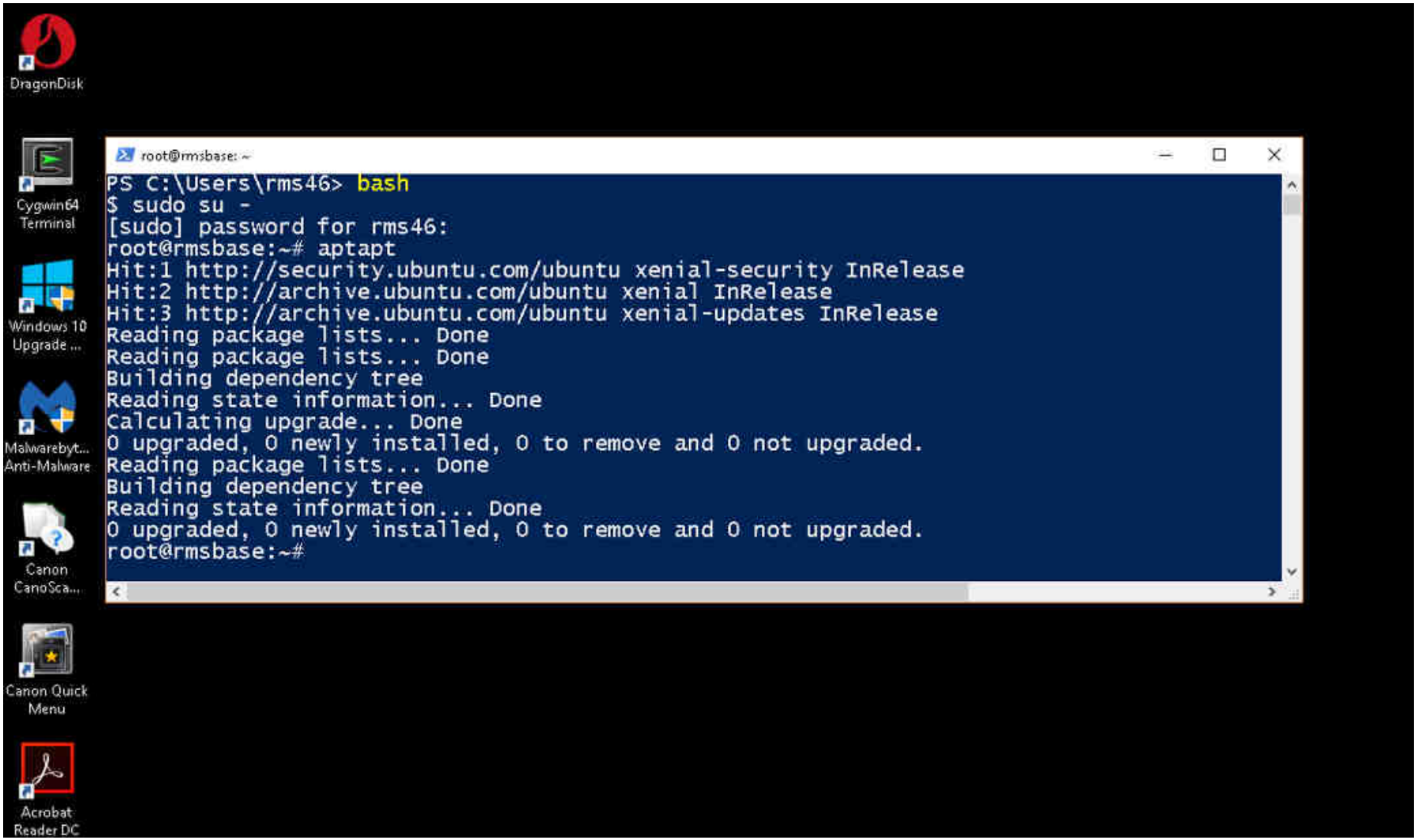


Figure: WSL: Windows Subsystem for Linux

Cygwin

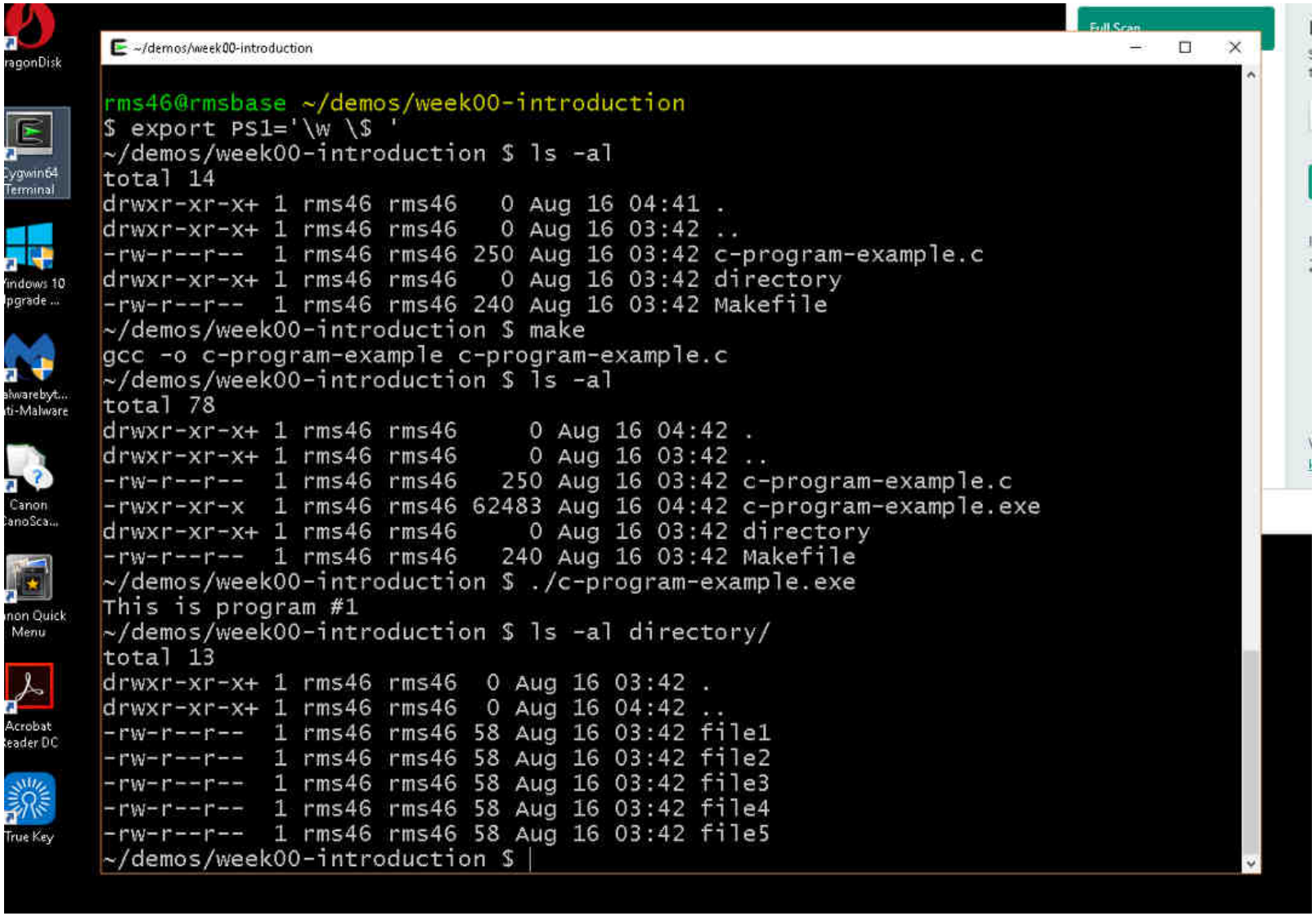


Figure: Cygwin

Program Example (Week 00)

```
$ 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

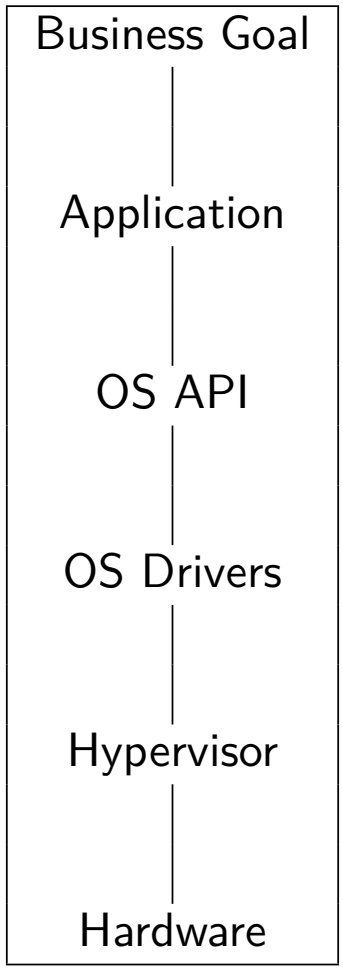
```
$ cat Makefile

# (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
```

Week 00: Demo Directory

```
$ 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
-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: Review

- What is an Operating System?
- Why taking an Operating System class?



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

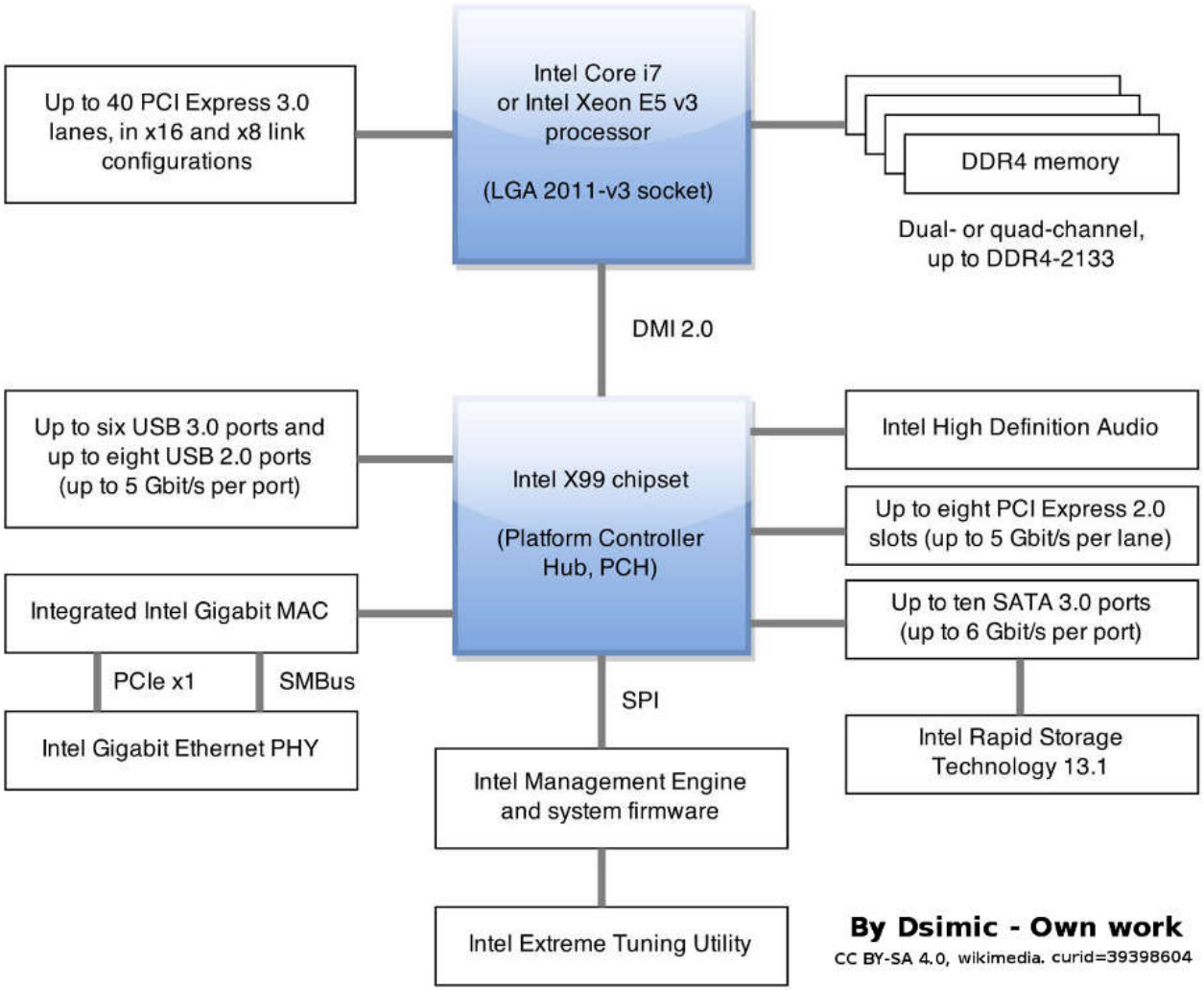


Figure: Block Diagram

APIC (Advanced Programmable Interrupt Controller)

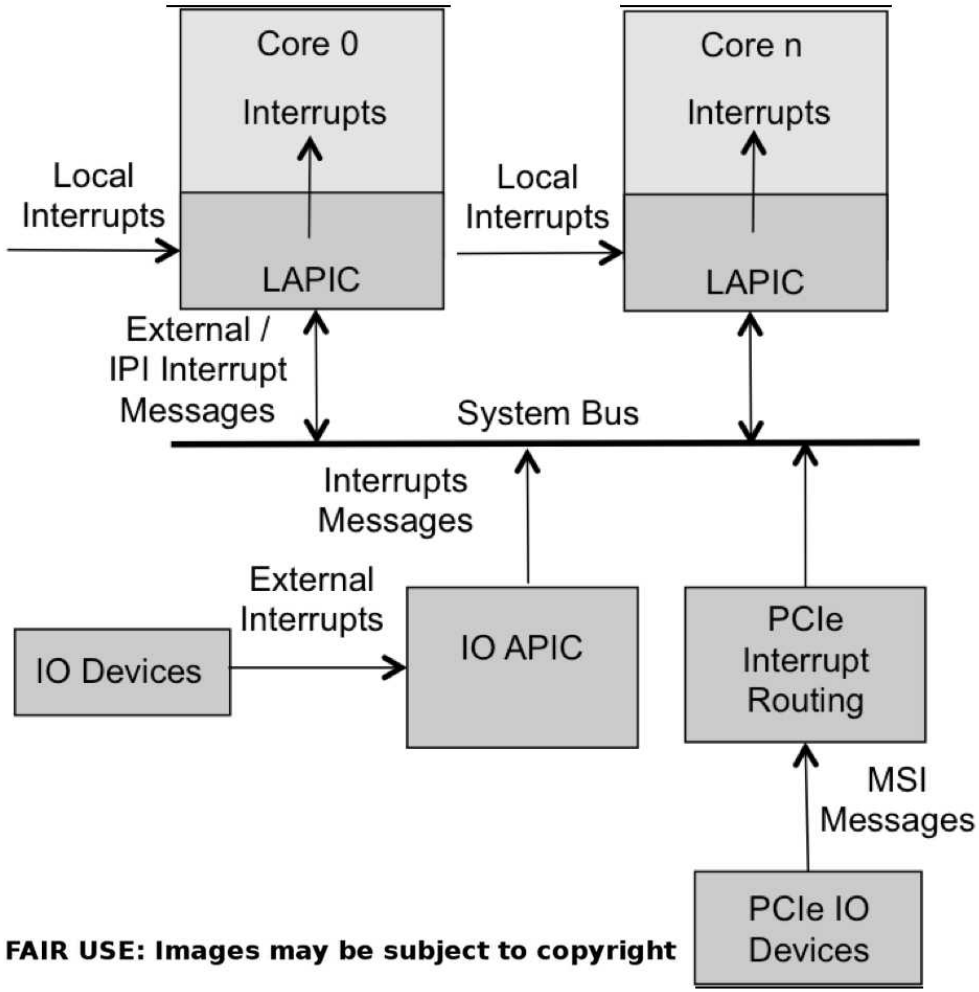


Figure: APIC (Advanced Programmable Interrupt Controller)

Interupt Handling

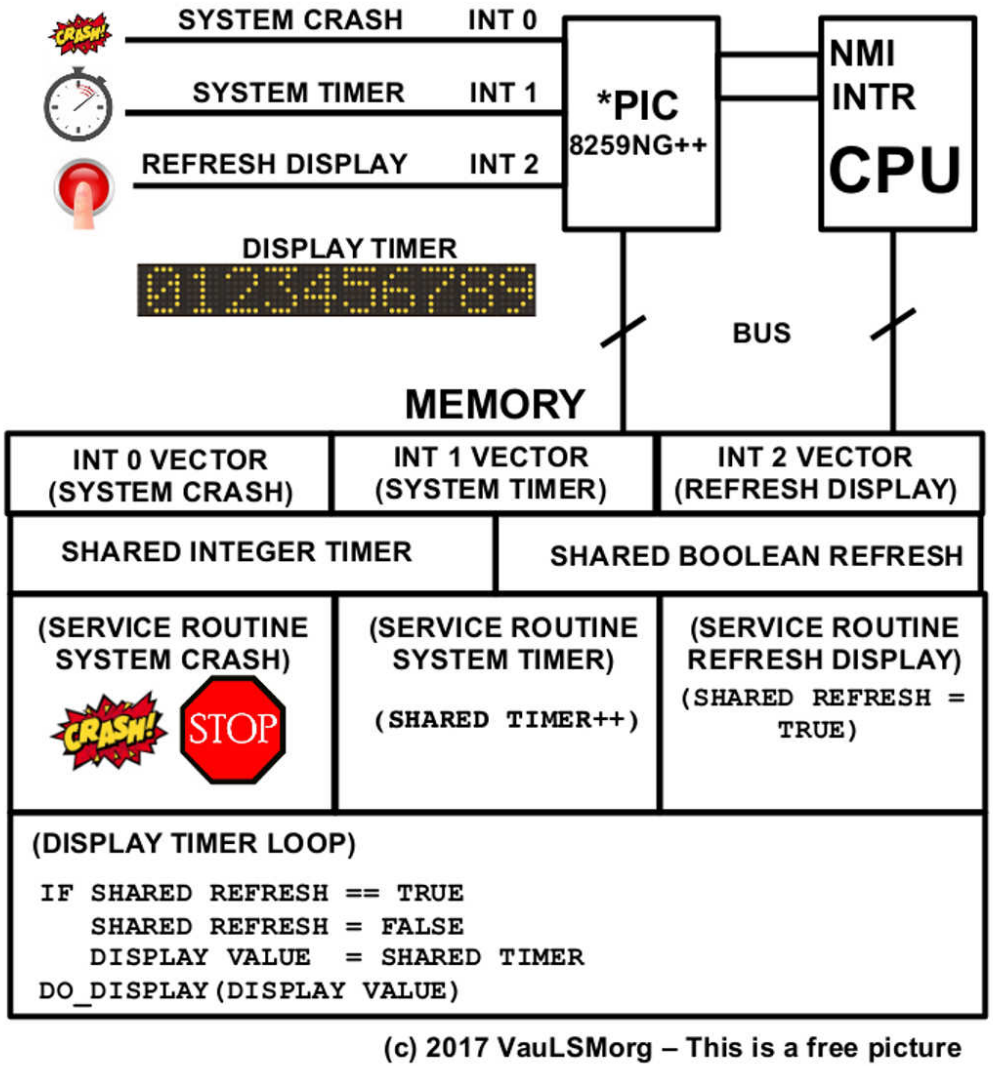
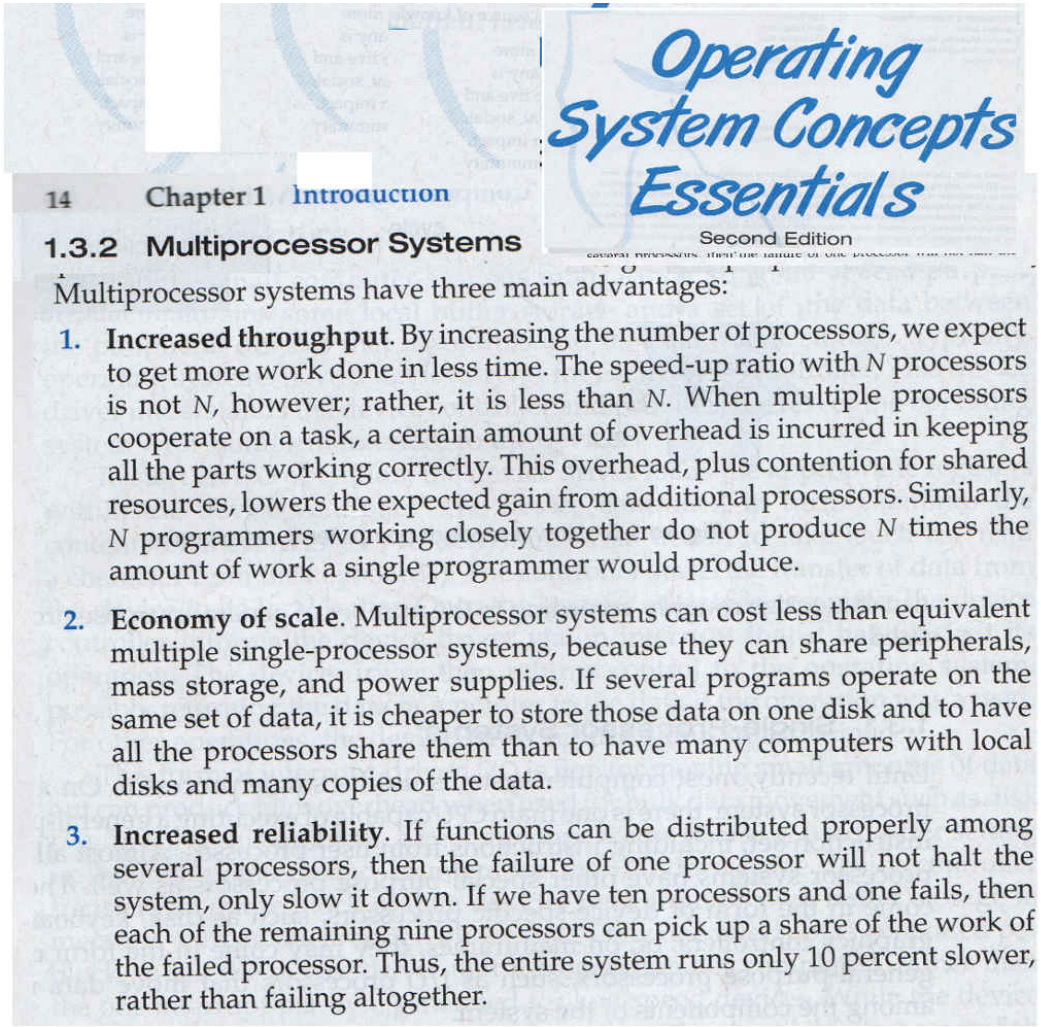


Figure: Interupt Handling with PIC (Programmable Interrupt Controller)

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.

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.

Week 00: Problems

- Tugas Minggu 00 (Week 00) ada dua:
  - membuat QRC dan mengirimkannya via email.
  - membuat Memo Minggu 00 yang ada QRC, serta ditunjukkan pada saat istirahat kuliah hari ke dua.
- "TANDA PETIK" BUKAN merupakan bagian dari QRC!
- Jangan mencantumkan ".git" dan ".sso", jika bukan bagian dari nama akun anda!
- Tanpa header [W00] pada Subject; email anda mungkin akan nyasar entah kemana... Ingat: [W00] (We-Nol-Nol) tidak sama dengan [WOO] (We-O-O)!
- Ukuran QRC cukup sekitar 256 x 256 pixel: jangan terlalu besar atau terlalu kecil.
- QRC ditanam (embedded) dalam email; jangan menggunakan attachment!
- Jangan mengirim MEMO dalam format PDF!

Week 00: Summary

- Reference: (OSC9 chapter 1 + chapter 16)
- What is an Operating Systems?
  - Definition: Resource Allocator & Control Program.
  - Why taking an Operating System class?
- Computer Organization Review
- The Manager Set
  - Process Manager, Memory Manager, I/O Manager, Storage Manager.
- Protection and Security
- Virtualization
  - Hypervisor type 0, 1, 2
  - Paravirtualization, Emulators, Containers.
  - VCPU: Virtual CPU
  - Virtualization Implementation:
    - Trap-and-Emulate mode
    - Binary Translation mode

Week 00: Check List

- ☐ Find/copy this document from `http://rms46.vlsm.org/2/207.html`
- ☐ Find/read a recent OS Book and map it to OSC9.
- ☐ Using your **SSO** account, login to `badak.cs.ui.ac.id` via `kawung.cs.ui.ac.id`.
- ☐ Check folder `badak:///extra/Week00/`
  - ☐ Try to copy and compile `c-program-example.c`.
- ☐ Create **public** project "os181" on your new (or existing) `github.com` account.
- ☐ Write in "README.md" file:
  "ZCZC Sistem Operasi 2018 Awal (1)"
- ☐ Encode your QRC.
- ☐ Mailto: `os181@vlsm.org` (Subject: [W00] CLASS ID SIAK-NAME)
- ☐ Write "Memo Week00" + your QRC.
- ☐ **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.