CSF2600505 Sistem Operasi
CSGE602055 Operating Systems
Minggu 00: Intro

Rahmat M. Samik-Ibrahim

Universitas Indonesia

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

REV54 27-Aug-2017

Agenda

1. Start
2. Agenda
3. Operating Systems
4. TOP 10 OS
5. Goal
6. Assessment
7. Resources
8. Schedule
9. Week 00: Introduction
10. Review
11. Managers Set
12. Potpourri
13. Lab and QR Code
14. Tools
15. Github Demo
16. Lab Programing
17. The End

Rahmat M. Samik-Ibrahim

- UI: sejak 1984.
- 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: `facebook.com/RMS46F/`
- Kontak: WhatsApp?

TOP 10 OS

1. Nama saya **Rahmat**. Rahmat nama saya. Kalau bukan Rahmat, bukan nama saya!
2. Jangan datang lebih lambat dari pada Pengajar!
3. Jangan berisik/asyik sendiri dalam kelas!
4. Jangan lupa mempersiapkan diri untuk berpartisipasi dalam kelas!
5. Jangan lupa membawa selembar kertas (+QRC) untuk membuat memo kuliah!
6. Jangan main "*games*" dan "*chat*" dengan "*gadget*" anda!
7. Jangan meminjam peralatan selama kuis dan ujian!
8. Jangan curang!
9. Jangan menghubungi Pengajar untuk masalah Administratip!
10. Jangan menjadi "*Puss in Boot*"!

Jangan menjadi Puss In Boot

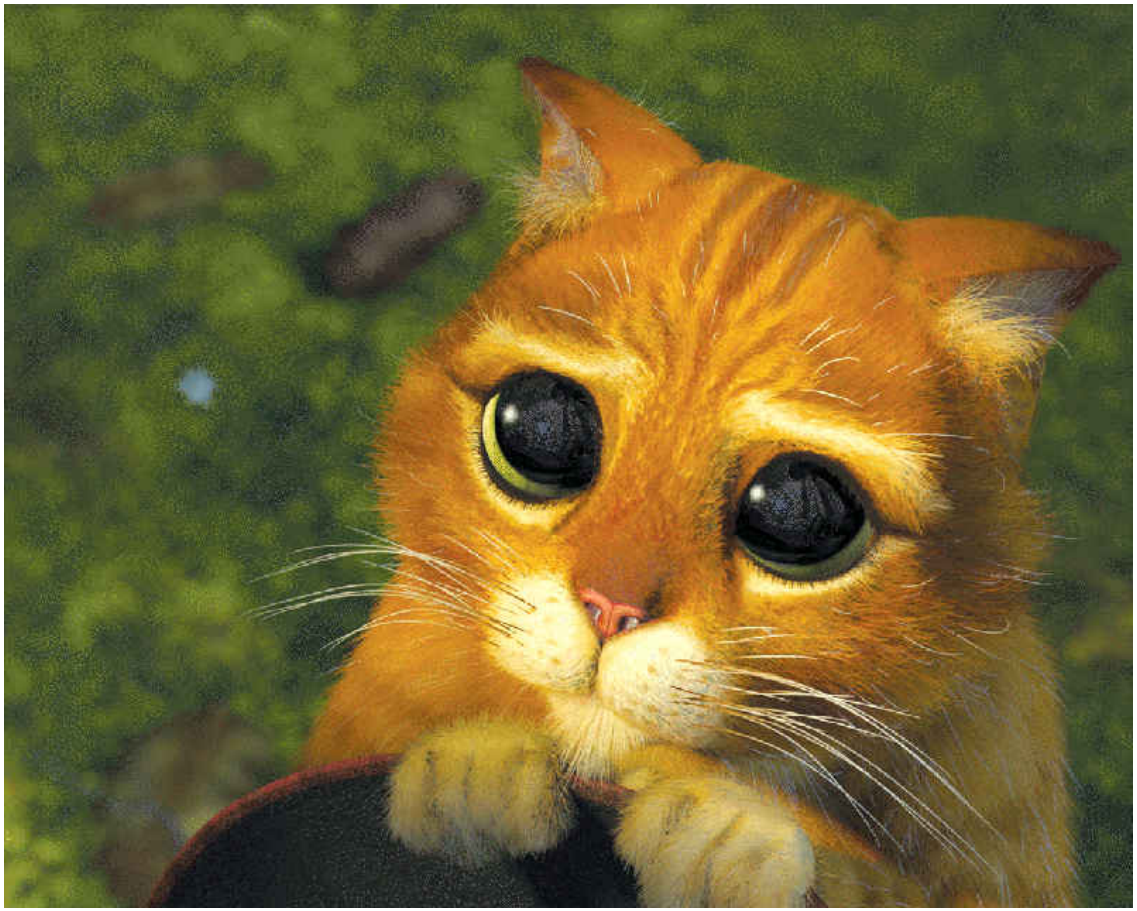


Figure: Ini Puss in Boot[1].

[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.

ETC

- **4 SKS**: Alokasikan 12 jam per minggu
- **No Lab. Assistant, No Teaching Assistant: Graders only**.
- Harap menghubungi SEKRE (Gedung B lantai 2) untuk segala masalah administratip, terutama absen, sakit, surat sakit, ujian susulan, dst.
- Harap merampungkan masalah administrasi ujian susulan dalam 6 hari kerja.

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]

- 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].
- Silakan membawa kertas A4 ke ruang ujian[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: "01110010"[1]
- ARCHIVE (Arsip bahan pengajaran): **https://scele.cs.ui.ac.id/course/view.php?id=126**
  - Enrollment key: "11010010"[1].

[1]Sewaktu-waktu akan diganti! Harap pantau "Announcement" secara teratur.

Schedule part 1

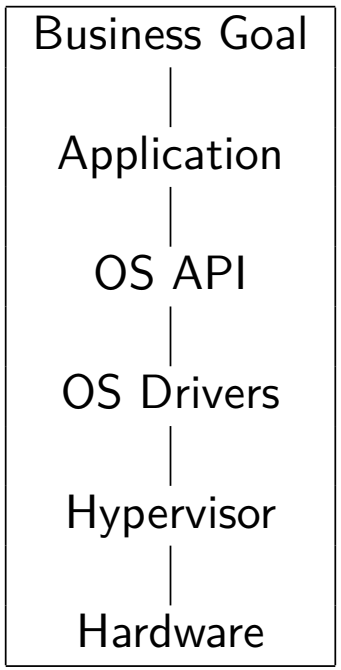
- Week00 – Intro (OSCE2e ch1/2)(UCB 01)(UDA P1L1/2) (OLD 00)
- Week01 – IPR & Scripting (ETC 000 001 002)(OLD 02-HKI 02-scripting) (Any Related Tutorial)
- Week02 – Protection, Security, & C-language (OSCE2e ch13-4) (ETC 050-1 C001-8) (OLD 01) (Any C Language Tutorial)
- Week03 – BIOS, Boot, & Systemd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

Schedule part 2

- Week06 – 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 & Sockets (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 – Cloud System & Virtualization (UCB 24) (SUP WEEK10)
- UAS 06 07 08 09 10

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.
  - 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: $9AB\ CDEF_{16} = 1001\ 1010\ 1011\ \ 1100\ 1101\ 1110\ 1111_2$

Managers Set

- Process:
  - Creating/Deleting; Suspending/Resuming; Synchronization; Communication;
- 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 & Schedulling:
  - Protecting.
  - Schedulling.

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.

QR Code: OS172 CLASS ID NAME

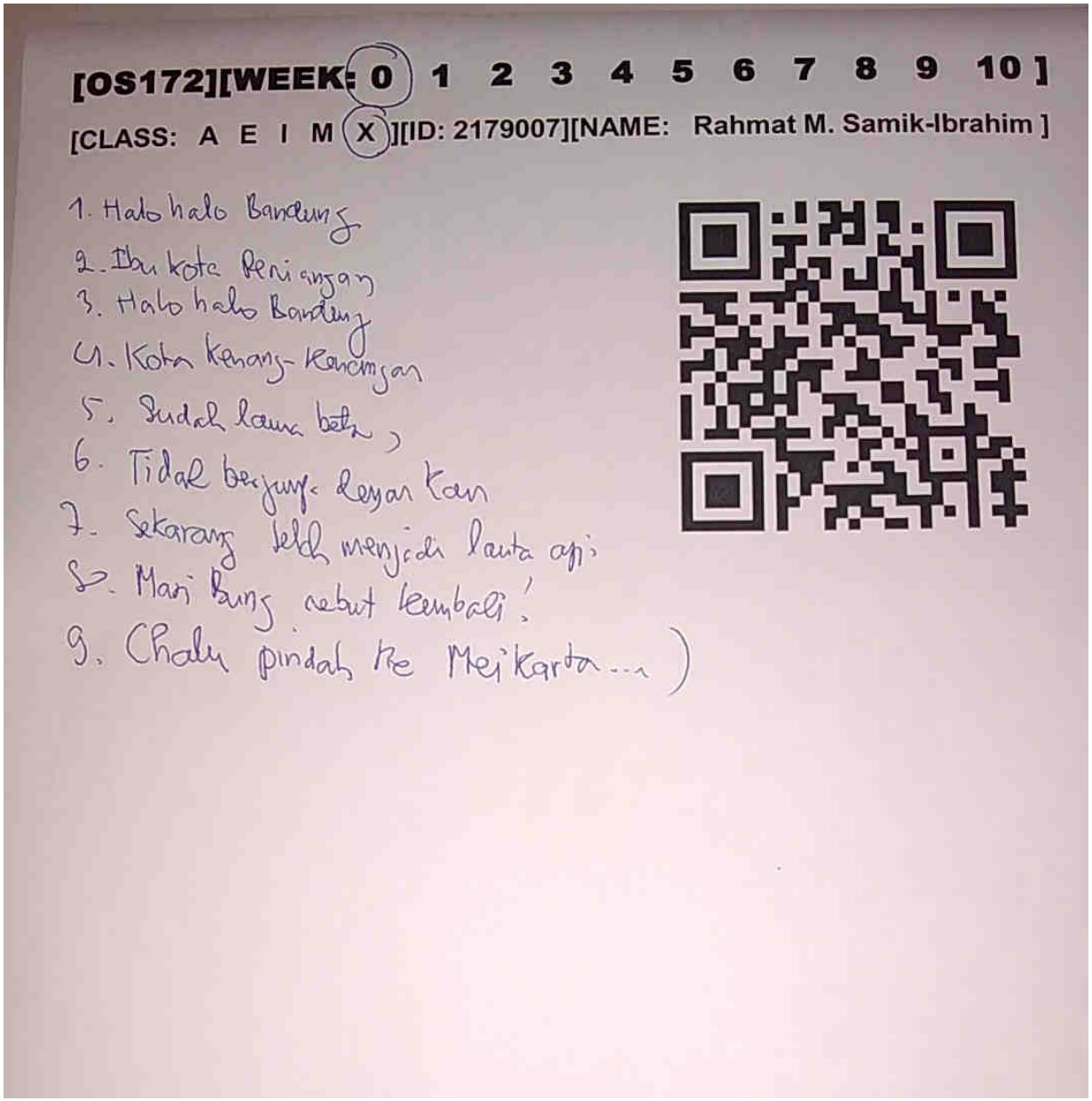
```
# OS172: OS 2017 2nd term
# CLASS: A (reguler) E (Extention) I (International) M (Matriculation)
# ID:    Student ID (NPM)
# NAME:  NAME (SIAK)

$ qrencode -o - -s 20 \
  "OS172 A 2179009 Rahmat M. Samik-Ibrahim" \
  | convert - -quality 25 OS172-A-2179007.jpg

$ zbarimg OS172-A-2179007.jpg
QR-Code:OS172 A 2179009 Rahmat M. Samik-Ibrahim
scanned 1 barcode symbols from 1 images in 0.21 seconds
```

Quick Response (QR) Code

Quick Response Code



Self Service Lab

- Github Account
  - `https://github.com/UI-FASILKOM-OS/os172`
  - `https://github.com/UI-FASILKOM-OS/demo`
- BASH Account:
  - Virtual Ubuntu: badak.cs.ui.ac.id (SSO)
  - Ubuntu (BYOD)
  - WSL: Windows 10 Subsystem for Linux
  - Cygwin (Windows)

WSL 1

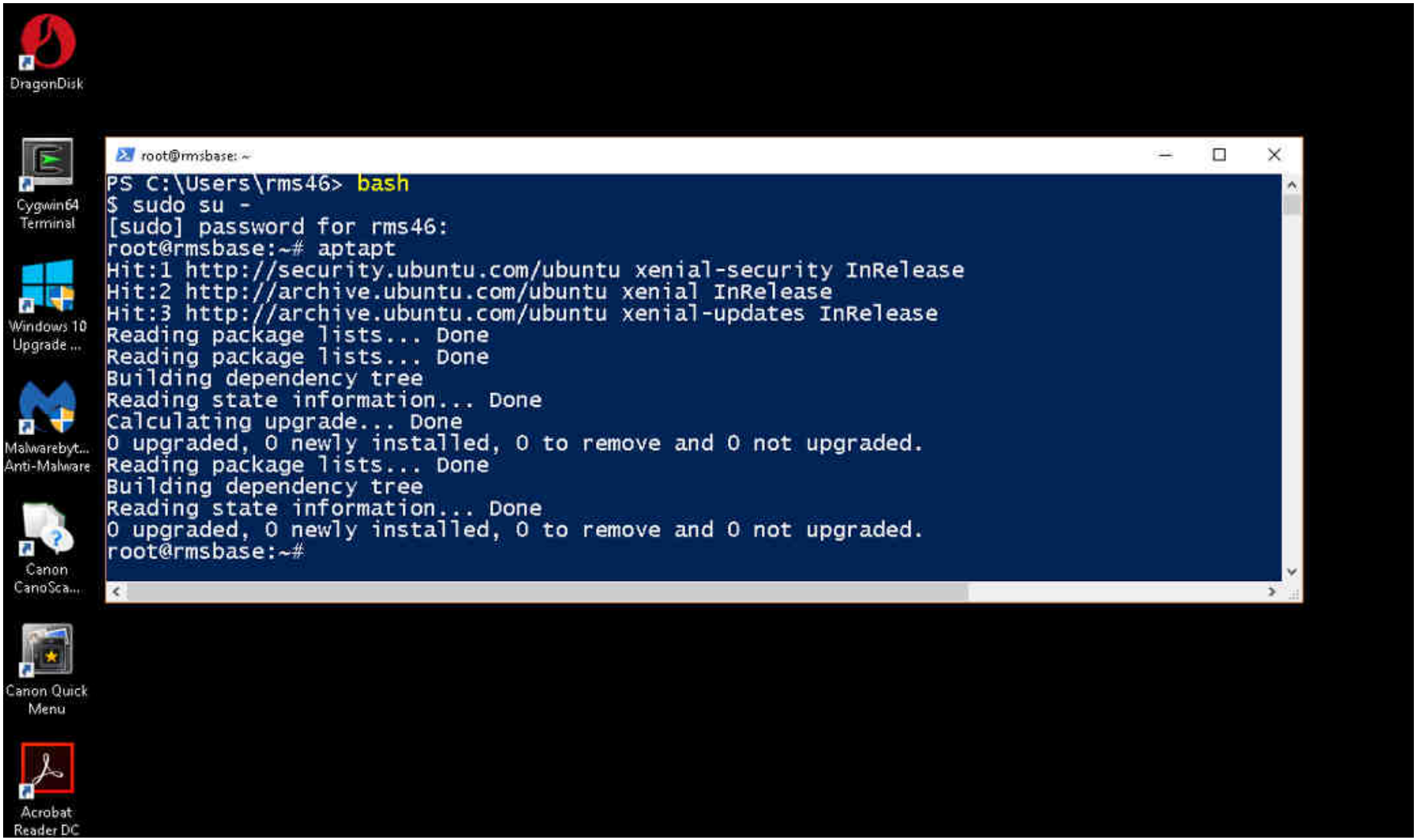


Figure: WSL: Windows Subsystem for Linux

WSL 2

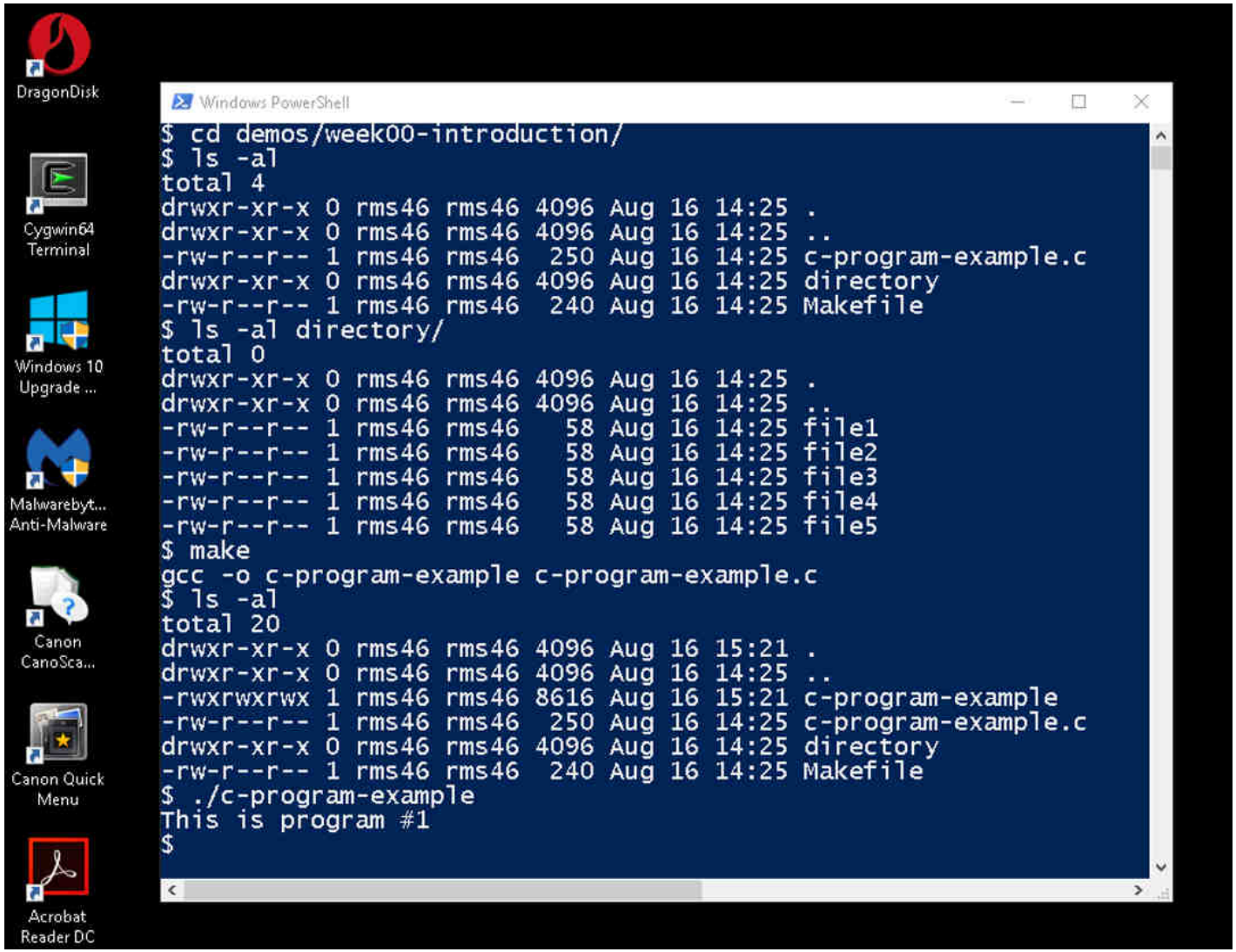


Figure: WSL: Windows Subsystem for Linux

Cygwin

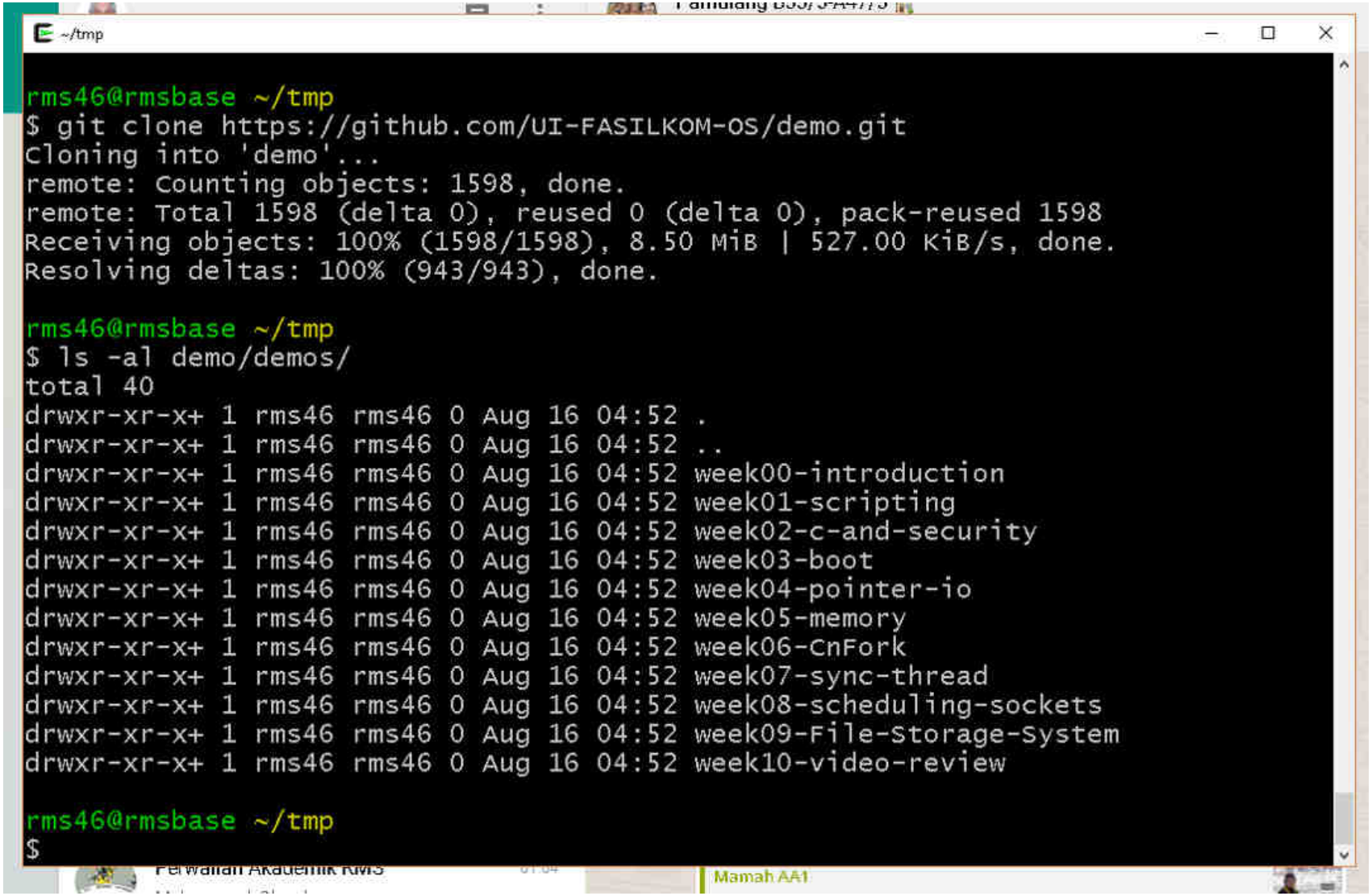


Figure: Cygwin

Cygwin

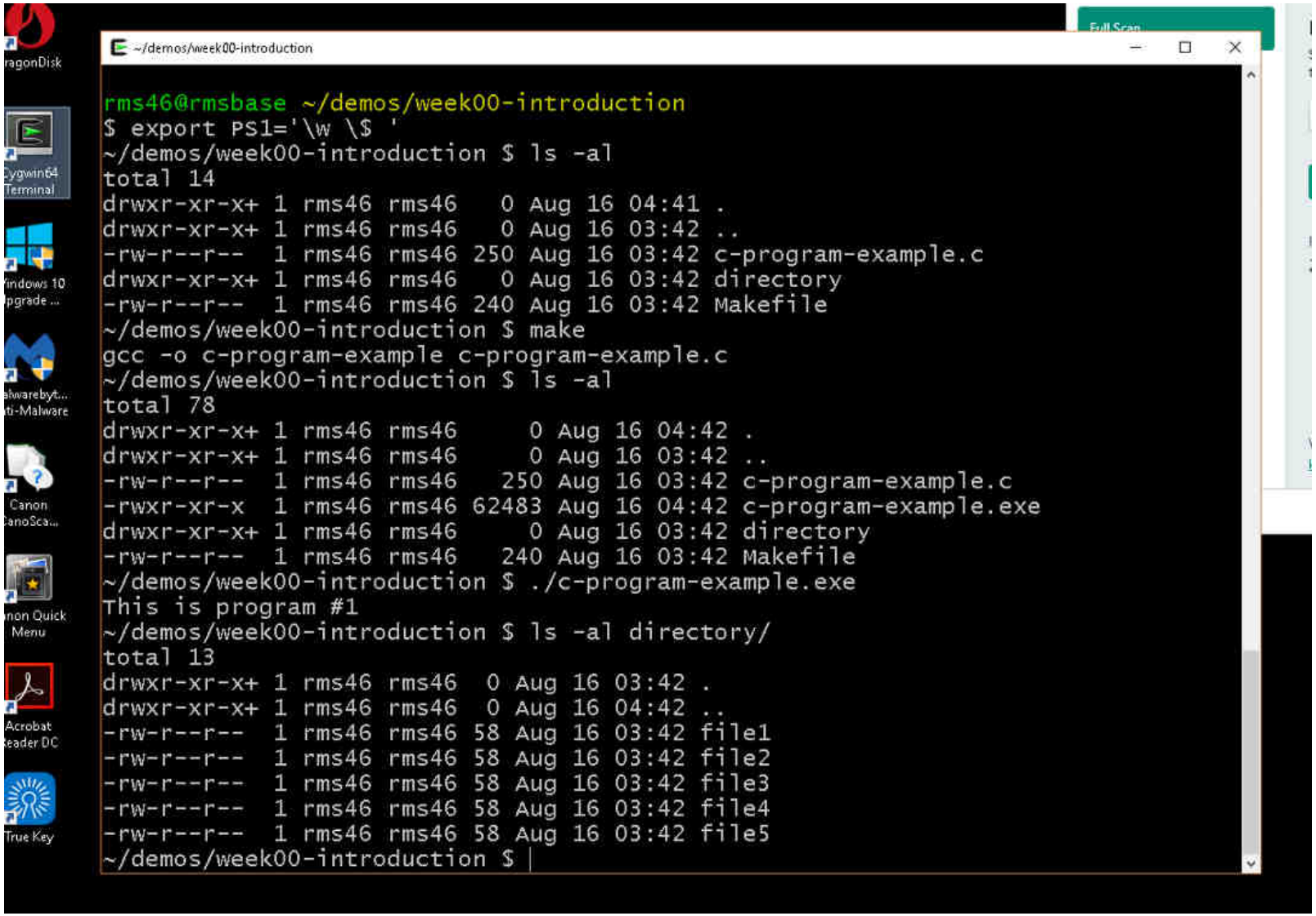


Figure: Cygwin

Github Demo

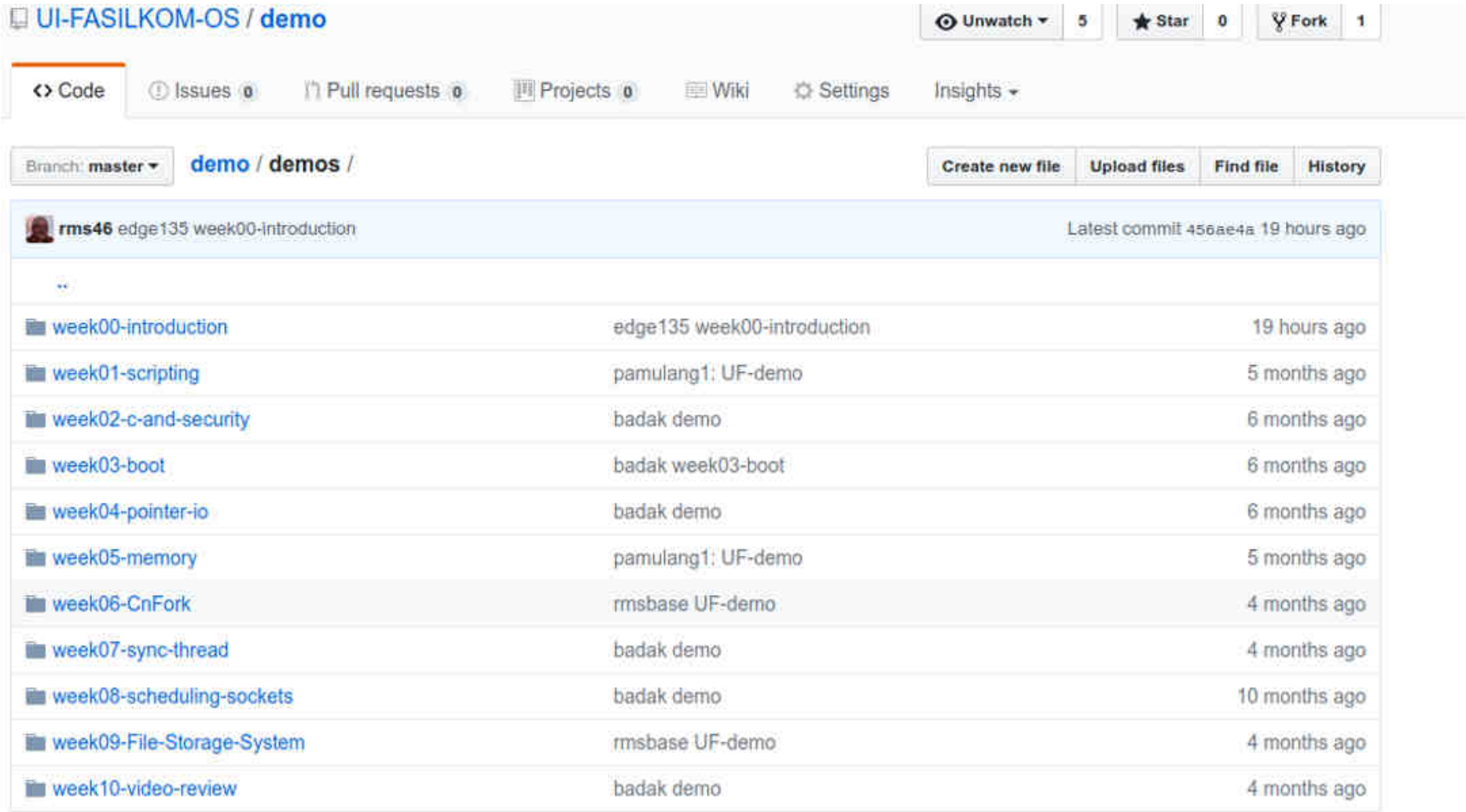


Figure: `https://github.com/UI-FASILKOM-OS/demo`

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
```

The End

- This is the end of the presentation.