FIT2100 Semester 2 2019
Operating Systems:
Unit Introduction

WEEK 1 PART A

What this unit is about

- How an OS works
- How to write native software that works with it

What are you going to learn?

- Topics covered: operating system structures and services, multi-programming processes, CPU scheduling, synchronisation algorithms, deadlocks, memory management approaches, device management, file management, operating system security, system calls and MORE!

Upon the completion of this unit, you should be able to:

1. Analyse and evaluate various strategies used by an operating system in managing the system resources and running applications efficiently;

2. Analyse and identify parameters that can improve the performance of multi-programming operating systems;

3. Apply synchronisation techniques in the development of applications, using operating system services; and

4. Apply system calls in the design of applications, to improve application performance, robustness or functionality.

FIT2100 Semester 2 2019 Unit Schedule

Week	Activities	Assessment
1	Computer Systems Overview	**Tutorials commence in Week 1** (alternate weeks)
2	Operating Systems Overview Design considerations	**Practicals commence in Week 2** (alternate weeks)
3	External I/O and file storage devices Hard disk access scheduling	
4	Filesystems System calls to enable file I/O by applications	
5	Processes and process management	**Programming Assignment Part A:** Due on 30 August 2019 (Friday) 5pm
6	Uniprocessor scheduling	
7	[NO LECTURES]	**Mid-semester test**: held during lecture times. **Programming Assignment Part A:** Interview (After tutorial session)
8	Concurrency part 1 Threads Mutual exclusion and synchronisation	
9	Concurrency part 2 Deadlocks Starvation	
10	Memory Management and Virtual Memory Shared memory	**Programming Assignment Part B:** Due on 11 October 2019 (Friday) 5pm
11	Interprocess communication	**Programming Assignment Part B:** Interview (After tutorial session)
12	Operating System Security Review	

FIT2100 Semester 2 2018 Assessments

- In-semester assessments:
  - 50% of the overall unit mark
  - Practical and tutorial assessment (10%)
  - Two programming assignments (30%):
    - Part A (15%) due in Week 5
    - Part B (15%) due in Week 10
  - Unit test (10%) during Week 7 lectures

- Examination:
  - 50% of the overall unit mark
  - 2-hour written exam
  - + 10 minutes reading time

FIT2100 Unit Hurdles

❑ To pass a unit, you must obtain:

- 40% or more of the exam mark
- 40% or more of the total mark for assignments and mid-sem test
- An overall unit mark of 50% or more

❑ If you do not pass any of these hurdles, and your overall unit mark is:

- equal to or greater than 50%, then a mark of 49 N will be recorded
- less than 50%, then the actual mark will be recorded

FIT2100 Tutorials and Practicals

❑ Tutorials (2 formal hours + 1 hour optional consultation):

- Weeks 1, 3, 5, 7, 10, 11

- Focus on the concepts and skills of C programming language

❑ Practicals (3 formal hours):

- Weeks 2, 4, 6, 8, 9, 12

- Focus on hands-on experience of operating system concepts

❑ Attendance for both tutorials and practicals are compulsory

❑ Please come prepared by completing the pre-class activities

FIT2100 Semester 2 2019 Teaching Staff

❑ Lecturer (Clayton Campus, Victoria):

- Mr Daniel Kos
- Email: Daniel.Kos@monash.edu

❑ Lecturer (Sunway (KL), Malaysia):

- Dr. Lim Chern Hong
- Email: Lim.Chernhong@monash.edu
- Office: 2414, School of Information Technology, Malaysia

❑ Lecturer (IIE MSA, Johannesburg, South Africa):

- Yolanda Kanyama

FIT2100 Semester 2 2018 Teaching Staff

❑ Tutors (Clayton):

- **Sami Samiullah (Sami.Samiullah@monash.edu) [Student engagement]**
- **Guy Kijthaweesinpoon (Guy.Kijthaweesinpoon@monash.edu) [Administrative matters / Special consideration requests]**
- **Daniel Jitnah (Daniel.Jitnah@monash.edu) [Linux systems]**
- Akshay Sapra (Akshay.Sapra@monash.edu)
- Pasindu Epa (Pasindu.Epa@monash.edu)
- Nick Chen (nick.chen@monash.edu)
- Anuj Nepal (anuj.nepal@monash.edu)
- Norman Chen (TBA)

❑ Tutor (Malaysia):

- Ms. Najini Arachichige (najini.arachichige@monash.edu)

  Additional consultations will be announced on Moodle.

Learning resources

❑ Moodle site (go to: lms.monash.edu)
- Lecture notes
- Tutorial and practical notes
- Lecture live stream and recordings
- Student forums
- **Unit announcements (important info will be communicated here).**
  - Subscribe to get these announcements by email

❑ Additional resources on Moodle (under Unit Information)
- eBooks
- Linux Virtual Machine image
- Setup guide for Linux Virtual Machine

MONASH University

❑ William Stallings (2017). Operating Systems: Internals and Design Principles (9th Edition), Pearson.

- Generally, each lecture corresponds to a chapter in this textbook
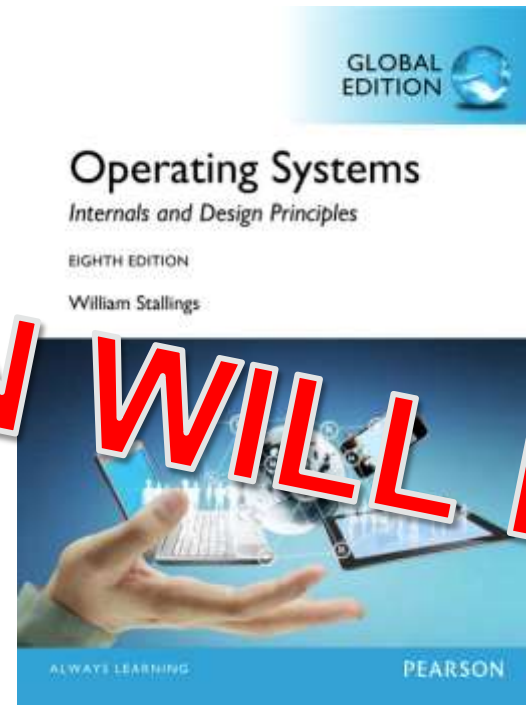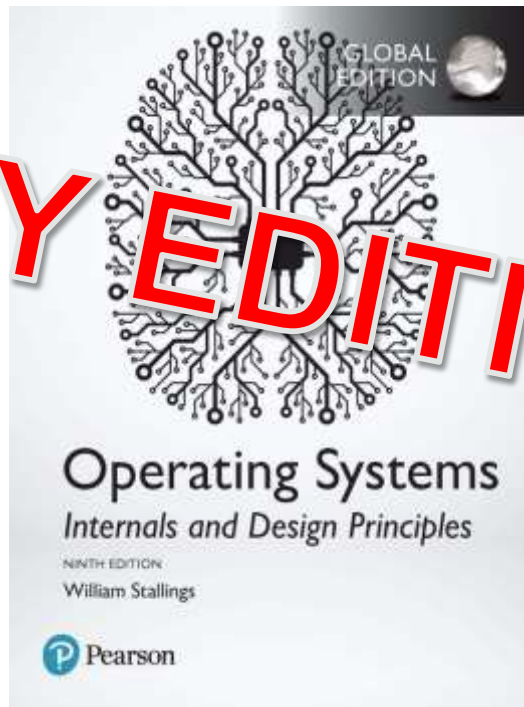
Other texts (Operating Systems concepts)

- ❑ Abraham Silberschatz, Peter B. Galvin, and Greg Gagne (2018). Operating System Concepts (10th Edition), Wiley.

- ❑ Abraham Silberschatz, Peter B. Galvin, and Greg Gagne (2013). Operating System Concepts (9th Edition), Wiley.

- ❑ Andrew S. Tanenbaum, and Herbert Bos (2014). Modern Operating System (4th Edition), Pearson.

FIT2100 Recommended Texts (C Programming for Unix)

❑ Brian W. Kernighan, Dennis M. Ritchie (1988). The C Programming Language (2nd Edition), Prentice Hall.

  • Official guide by the original creators of C

❑ K. N. King (2008). C Programming: A Modern Approach (Second Edition), W. W. Norton.

❑ Ben Klemens (2014). 21st Century C: C Tips from the New School (2nd Edition), O'Reilly.

❑ David A. Curry (1989). C on the UNIX System (Internet Edition), O'Reilly.

❑ David A. Curry (1996). UNIX Systems Programming for SVR4 (Internet Edition), O'Reilly.

❑ W. Richard Stevens, and Stephen A. Rago (2013). Advanced Programming in the Unix Environment (Third Edition), Addison-Wesley.

MONASH University

THIS UNIT IS IMPORTANT…

❑ This unit will give you skills and knowledge you won't obtain elsewhere

- A **deeper** understanding of systems-level programming makes you more valuable in industry
- You will see simple solutions to tricky problems

❑ High-level programming environments come and go like fads

- If you cling to the comfort of a particular coding environment, you will be at the mercy of a changing world!

❑ Underlying low-level concepts are valid for **decades**

- You will be able to adapt to a changing industry
- You will be able to adapt to new programming systems based on a deeper understanding

❑ The next slide might make you feel stressed

OUR JOB IS TO TAKE YOU OUT OF YOUR COMFORT ZONE!

…YOUR ATTITUDE IS *MORE* IMPORTANT

❑ Old assumptions are shattered as you gain new knowledge

❑ Engage *with* the material, not against it!

- Seek to *challenge* your current perspective – don't just seek validation
- Negative engagement is not only self-destructive to your learning experience, it can demoralise your peers

❑ Do these thoughts sound familiar?

- *'This unit would make more sense if they taught it my way…'*
- *'I was smarter before I took this unit. This unit made me stupid!'*
- *'I'm already an expert and I haven't learned anything new this semester. Why didn't I get an HD?'*

MONASH University

AN UNPRODUCTIVE FORUM POST…

❑ Why is this forum exchange problematic? What would be a more useful engagement approach?

*Dear teaching team, I would like to give my **student feedback** on the assignment which is due tomorrow. I feel the assignment is too much work because we are not told **how** to solve the problem. It is **unfair** not to give us a more detailed marking scheme so we can simplify our work by ticking off the code structures that our finished assignment should contain to get all of the marks!*

*I would also like to second this. It is also annoying that this unit does not even let us use the same **familiar** programming environment students are already comfortable with and…*

Plagiarism and Collusion

❑ Student responsibilities:

- Undertake studies and research responsibly and with honesty and integrity
- Ensure that academic work is in no way falsified
- Seek permission to use the work of others, where required
- Acknowledge the work of others and cite references used
- Take reasonable steps to ensure *other* students are not able to copy or misuse your work

❑ Student Academic Integrity Policy:

- http://www.monash.edu.au/students/policies/academic-integrity.html

Disability Support Services

Do you have a disability, medical or mental health condition that may impact on your study?

Disability Support Services (DSS) provides a range of services for registered students including:

- Note takers and Auslan interpreters
- Readings in alternative formats
- Adaptive equipment and software
- Alternative arrangements for exam and class tests

DSS also support students who are *carers* of a person with a disability, medical or mental health condition, or requiring aged care.

For further information and details about how to register:

- T: 03 9905 5704
- E: disabilitysupportservices@monash.edu
- W: www.monash.edu/disability