



COMP3231/9201/3891/9283

Operating Systems 2021/T1

UNSW

Administration

- [Notices](#)
- [Course Outline](#)
- [UNSW Timetable](#)
- [Consultations](#)
- [Survey Results!!](#)

Work

- [Lectures](#)
- [Tutorials](#)

Support

- [Ed Forums](#)
- [Wiki](#)

Assignments

- [Submission Guide](#)
- [Assignment 0 Warm-up](#)

Resources

OS/161

- [General](#)
- [Man Pages](#)
- [Sys161 Pages](#)

C coding

- [Info Sheet](#)

Debugging

- [Learn Debugging](#)

General

- ["Hardware" Guide](#)
- [R3000 Reference Manual](#)
- [Intro. to Prog. Threads](#)

Previous years

- [2020 T2](#)
- [2020 T1](#)
- [2019 T1](#)
- [2018 S1](#)
- [2017 S1](#)
- [2016 S1](#)
- [2015 S1](#)
- [2014 S1](#)
- [2013 S1](#)
- [2012 S1](#)
- [2011 S1](#)
- [2010 S1](#)
- [2009 S1](#)
- [2008 S1](#)
- [2007 S1](#)
- [2006 S1](#)
- [2005 S2](#)
- [2005 S1](#)
- [2004 S2](#)
- [2004 S1](#)

Tutorial Week 5

Questions

Memory Hierarchy and Caching

1. Describe the memory hierarchy. What types of memory appear in it? What are the characteristics of the memory as one moves through the hierarchy? How can do memory hierarchies provide both fast access times and large capacity?
2. Given that disks can stream data quite fast (1 block in tens of microseconds), why are average access times for a block in milliseconds?
3. You have a choice of buying a 3 Ghz processor with 512K cache, and a 2 GHz processor (of the same type) with a 3 MB cache for the same price. Assuming memory is the same speed in both machines and is much less than 2GHz (say 400MHz). Which would you purchase and why? Hint: You should consider what applications you expect to run on the machine.

Files and file systems

4. Consider a file currently consisting of 100 records of 400 bytes. The filesystem uses *fixed blocking*, i.e. one 400 byte record is stored per 512 byte block. Assume that the file control block (and the index block, in the case of indexed allocation) is already in memory. Calculate how many disk I/O operations are required for contiguous, linked, and indexed (single-level) allocation strategies, if, for one record, the following conditions hold. In the contiguous-allocation

Staff

- [Kevin Elphinstone \(LiC\)](#)
- TBD (Admin)

Grievances

- [Student Reps](#)



case, assume that there is no room to grow at the beginning, but there is room to grow at the end of the file. Assume that the record information to be added is stored in memory.

- The record is added at the beginning.
- The record is added in the middle.
- The record is added at the end.
- The record is removed from the beginning.
- The record is removed from the middle.
- The record is removed from the end.

5. In the previous example, only 400 bytes is stored in each 512 byte block. Is this wasted space due to internal or external fragmentation?

6. Old versions of UNIX allowed you to write to directories. Newer ones do not even allow the superuser to write to them? Why? Note that many unices allow you read directories.

7. Given a file which varies in size from 4KiB to 4MiB, which of the three allocation schemes (*contiguous, linked-list, or i-node based*) would be suitable to store such a file? If the file is access randomly, how would that influence the suitability of the three schemes?

8. Why is there VFS Layer in Unix?

9. How does choice of block size affect file system performance. You should consider both sequential and random access.

10. Is the `open()` system call in UNIX essential? What would be the consequence of not having it?

11. Some operating system provide a *rename* system call to give a file a new name. What would be different compared to the approach of simply copying the file to a new name and then deleting the original file?

12. In both UNIX and Windows, random file access is performed by having a special system call that moves the *current position* in the file so the subsequent `read` or `write` is performed from the new position. What would be the consequence of not having such a call. How could random access be supported by alternative means?

Page last modified: 4:49pm on Saturday, 23rd of May, 2020

[Print Version](#)

CRICOS Provider Number: 00098G