# csci3411: Written Homework for Practice

*Operating Systems*
Gabriel Parmer
**You will not submit this homework for credit.**

## 1 Instructions

These questions will take you out of your comfort zone, and will force you to think about the issues discussed in class and in the book. *Critical thinking*, though difficult, is good for you.

I strongly encourage you to discuss general concepts (e.g. what a monitor is) with other students, but your answer must be your own. Do not discuss specific questions with other students.

Please note any assumptions that you have to make.

Please remember that although no questions about virtual memory and protection hardware exist on this test, they will certainly be tested on the final.

### Question 1: Directory Structures

We have discussed both single-level (*flat*) and hierarchical file-system directory structures as mechanisms for naming files.

Could you simulate a hierarchical directory structure with a flat directory if you could assume arbitrarily long file names? How does this compare with a truly hierarchical structure, e.g. how many directory entries would need to be read when a `ls /foo` command is issued.

### Question 2: File Deletion

Imagine a system where the disk space for a file could be deallocated and another file stored in its space, all while a program maintains a link to the file. What problems could this cause? How is this problem solved in UNIX-like systems?

### Question 3: File-System Access

Consider a file `/a/b/c/foo` that is 5 blocks in length. Directory information (i.e. a list of subdirectories) is stored on disk in the same manner as file data. For the following questions, consider (a) contiguous, (b) linked, (c) single-level index, and (d) 2-level index allocation schemes. Additionally, assume no caching, and few subdirectories in `a`, `b`, and `c`. State any assumptions you make.

1) How many disk accesses are required to read the first block of the file?
2) How many disk accesses are required to read the fifth block of the file?

### Question 4: Extents

File-systems can be modified to use extents. For the following questions, consider (a) linked, and (b) 2-level index allocation schemes. How would you have to modify the data-structures and algorithms of these schemes to find a block in the middle of a file when either

1) extents are fixed in size (called clusters), or
2) extents can be of varying sizes.

How would these alterations effect the number of blocks accessed to retrieve the *nth* block of a file. How would these schemes effect fragmentation (internal and external).

### Question 5: Disk Scheduling

A disk has 5000 cylinders, numbered 0 to 4999. The disk head is currently at cylinder 150, and the previous request was at 126. Assume all requests arrived at the same instant. The queue of pending requests (in FIFO order) is:

100, 1475, 900, 1775, 950, 1500, 1025, 1750, 125

Starting from 150, what is the total distance the arm must move to satisfy all pending requests? Answer for:

1) FCFS
2) SSTF
3) SCAN
4) C-SCAN
5) LOOK
6) C-LOOK

For this workload, which algorithm provides the best throughput, and which provides the best fairness?

### Question 6: Access Control Mechanism and Policy

A system contains three users, $A$, $B$, and $C$, and six documents $a_1$, $a_2$, $b_1$, $b_2$, $c_1$ and $c_2$. Each user can have read and/or write access to a document.

a) Write out the access control matrix for this system whereby permissions are determined by the Bell-LaPadula policy. Each user is at a different level of confidentiality shared with the corresponding lower-case documents (i.e. $A$ is in the same level as $a_1$ and $a_2$). Assume that each user has maximum permissions possible for each document given the policy.

b) $A$ is deemed a traitor, and we wish to remove his/her access to all documents. i) Assume the reference monitor for the system is implemented with ACLs. How many ACLs will we need to modify to remove access? ii) If the system used capabilities instead, would we have to change more or less access control data-structures.

### Question 7: Memory Management

Assume that a program makes the following sequence of memory management calls:

```
a = malloc(2K)
b = malloc(12K)
c = malloc(6K)
free(b)
d = malloc(11K)
e = malloc(17K)
```

```
free(a)
f = malloc(1K)
free(d)
free(e)
g = malloc(9K)
```

Assume that no space is taken up by "headers" to track the size of an allocation. What is the total internal and external fragmentation at the end of this sequence for the following policies:

a) First-Fit

b) Best-Fit

c) Buddy allocation – assuming the allocator starts out with 128K of memory, and the minimum allocation is 4k.

Additionally:

d) Which freelists for the buddy allocator have memory chunks in them at the end of these operations (i.e. which sizes of freelists are non-empty)? The allocator makes the same assumptions as in (c).