COMP3230 Principles of Operating Systems

Problem Set #3

Due date: 23:59 pm, November 30, 2023

Total 5 points

(version: 1.0)

Note: This assignment weights a total of 5 points in the final marks of this course. However, just like each process has its own private virtual address space, this assignment also has its own virtual score space of 100 points, which is mapped to 5 points of physical marks. The below points for each question are given in the said virtual score space.

Question 1 – **Deadlock** (30 points)

(This question is related to ILO 2c – "explain the underlying causes of concurrency and deadlock issues".)

Consider a version of the Dining Philosopher's problem where there are 5 Philosophers and 10 chopsticks. Each of the Philosophers needs three chopsticks before she will start to eat (Yes, three chopsticks, as they need one to help them think at the same time while eating). The chopsticks are placed in the middle of the table as a shared pool. Every Philosopher will return all of their chopsticks to the shared pool when done eating.

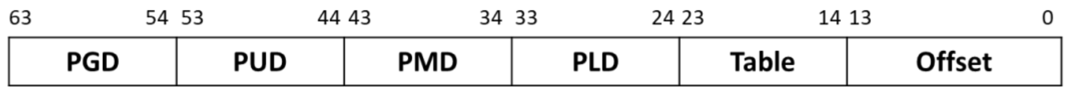
(1) Applying the idea of the Banker's algorithm, devise a simple rule for when it is safe for a Philosopher to pick up a chopstick (denote x as the number of chopsticks the Philosopher already holds). Explain why.

(2) Now suppose each Philosopher needs k chopsticks (k>3). Generalize the rule you developed above to work for any k. (Assume there are certainly more than k chopsticks in total, either on the table or held by some Philosophers).

Question 2 – Paging (50 points)

(This question is related to ILO 2b - describe the principles and techniques used by OS in effectively virtualizing memory resources.)

Consider a computer system with a 64-bit processor using a page size of 16 KB (16384 bytes) and a frame size of 16 KB. A junior OS designer plans to use a multi-level paging scheme with 5-level. They are the page global directory (PGD), page upper directory (PUD), page middle directory (PMD), page lower directory (PLD), and page table. For each level, it uses 10 bits of the virtual address to refer to information in each directory table or page table. To support a physical address space of more than 50 bits, each table entry of the directory/page table takes up 8 bytes. Here is the logical structure of his scheme.



(1) Being an experienced OS designer, you immediately notice that there is a shortcoming here. This scheme wastes a lot of precious memory when storing the directory and page tables.

i) Explain why this scheme suffers wastage of precious memories.
ii) Assume all page tables at the lowest level are placed in the main memory (i.e. ignore all directory tables), how many bytes of memory are wasted?

(2) Keep using a 5-level paging scheme, how should you redesign the logical structure of the virtual address such that you could reduce the memory wastage issue? You should clearly state the logical structure of your design and explain why it is better.

Question 3 – File System (20 points)

(This question is related to ILO 2d - describe the principles and techniques used by OS to support persistent data storage)

Consider a UNIX file system using inode to describe the metadata of a file. Assume the disk block is 4KB (4096 Bytes), and disk addresses are 4 bytes. Fill in your answers for below questions.

1) A directory in the file system is simply organized as a file, which contains a list of

   (___________________________________, _____________________________________)
   pairs for the files/directories in that directory.

2) Assume that an inode has 10 direct pointers, a single indirect pointer, and a double indirect pointer. The file can grow to be as big as ___________________________ bytes.

3) Suppose we want to open a file `/sapientia/et/virtus`, read it, and then close it. In doing so, the system will read (how many) _________________________________ inodes.

4) Now suppose `/motto/of/hku` is a ***symbolic link*** to `/sapientia/et/virtus`. (How many) ______________________________ inodes will be read in order to open it, read it, and then close it via `/motto/of/hku`.

5) Now assume the file `/sapientia/et/virtus` is opened, and a process appends a disk block of data to it 3230 times. Assume no buffering of writes. What data structures of the file will be modified during ***each*** append operation?
   _______________________________,________________________, ____________________.