United International University

Department of Computer Science and Engineering

CSI 309/CSE 4509: Operating System Concepts/Operating Systems

Final Examination: Spring 2024

Total Marks: 40 Time: 2 hours

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

Answer all the questions. Numbers to the right of the questions denote their marks.

1. Look at the code snippet below and answer the questions.

```
1   #include <bits/stdc++.h>
2   using namespace std;
3   #define NUM_OF_PERSON 5
4   #define THINK 5
5   bool forks[NUM_OF_PERSON]={1,1,1,1,1};
6
7   void acquire(bool *available){
8       while(*available==false);
9       *available = false;
10  }
11  void release(bool *available){
12      *available = true;
13  }
14  void philosopher_meetup(int i){
15      if(i%2){
16          acquire(&forks[i]);
17          sleep(THINK);
18          acquire(&forks[(i+1)%NUM_OF_PERSON]);
19          release(&forks[i]);
20          release(&forks[(i+1)%NUM_OF_PERSON]);
21          printf("When Synchronization meets Deadlock!\n");
22      }
23      else{
24          acquire(&forks[(i+1)%NUM_OF_PERSON]);
25          sleep(THINK);
26          acquire(&forks[i]);
27          release(&forks[i]);
28          release(&forks[(i+1)%NUM_OF_PERSON]);
29          printf("Are philosophers in deadlock?\n");
30      }
31  }
32  int main()
33  {
34      thread threads[NUM_OF_PERSON];
35      for (int i = 0; i < NUM_OF_PERSON; ++i) {
36          threads[i] = thread(philosopher_meetup,i);
37      }
38      for (int i = 0; i < NUM_OF_PERSON; ++i) {
39          threads[i].join();
40      }
41
42      return 0;
43  }
```

(a) **Draw** the resource allocation graph for the above code. [2]

(b) Is there a deadlock? If so, identify the philosopher IDs responsible for the deadlock. Otherwise, **determine** the output. [1]

(c) What will happen if we replace the statement of lines 25 and 27? [2]

2. Consider the Table-1 snapshot of a system:

Process	Allocation				Max			
	A	B	C	D	A	B	C	D
T0	3	0	1	4	5	1	1	7
T1	2	2	1	0	3	2	1	1
T2	3	1	2	1	3	3	2	1
T3	0	5	1	0	4	6	1	2
T4	4	2	1	2	6	3	2	5

Table 1: Process along with resources

(a) **Determine** whether the system is in a safe state or not if the available resources are {0,3,0,1}? If the state is safe, illustrate the order in which the threads may be completed. Otherwise,**Calculate** the minimum number of additional available resources required to ensure the execution of all threads. [2 + 1]

(b) **"If a request from thread T3 arrives for (0,0,0,1), the request will be granted immediately."- Justify** the statement. [**Assume, available resources={1,0,0,2}**] [2]

3. (a) If we have 1000 blocks on a disk, each with a block size of 128KB, and each block requires 10ns for access. Additionally, suppose every file is stored using a hash function without any collisions. If a file named "OS-final.pdf" with a memory requirement of 2KB is stored in a block of the disk, **Calculate** the maximum time required to access the file if accessing each byte requires 25ms? [2.5]

(b) **Convert** the File Allocation Table (FAT) represented in Figure-1 into linked allocation. [2.5]

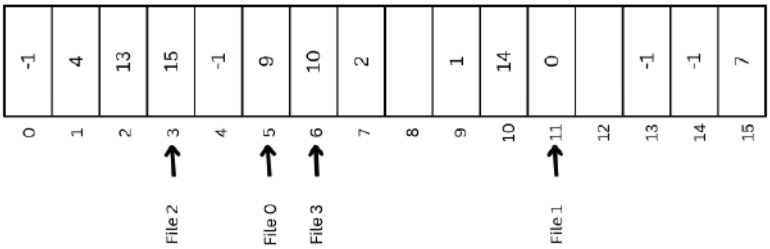

Figure 1: File Allocation Table

(c) Consider a system with a disk size of $8MB$ and a block size of $256KB$. Four files, **"index.html"**, **"style.css"**, **"NID.jpg"**, and **"bootstrap.css"** arrive sequentially to the CPU with memory requirements of $1.25MB$, $3MB$, $1.75MB$, and $1.5MB$, respectively. The CPU follows a contiguous allocation strategy, where after allocating each file, two blocks are left free before allocating another file.

i. Draw the contiguous allocation of disk space. [3]

ii. Will all the files be successfully allocated? If not, what would be the solution? [2]

4. Suppose, a system uses 4 GB physical memory and 64 GB of virtual memory. The system uses two different paging modes of 64KB and 1MB. The system is word addressable where each word is 4 bytes. A new process arrives at the system that uses the latter paging mode. The value of page table base register(PTBR) for that process is 400. The process executed the command **MOV REG1, 0x0002A8C6E**

(a) **Determine** the length of physical and virtual addresses. [0.5 + 0.5]

(b) How many bits are free for the meta information (valid/invalid bit and others) of the page table entries? [1]

(c) **Calculate** the size of the page table for the process. [2]

(d) **Find** the physical address where the process will find the required entry for address binding while executing the command. [1]

5. A system with 16-bit virtual address, 15-bit physical address, and a page offset of 4KB is running a program with the following assembly instructions. The system uses second-chance as their algorithm of choice for page replacement.

```
LEA    R1, 10002         ; Loads the value stored in the address 10002 to R1
PUSH   0x0C0C            ; Pushes the value at 0x0C0C on top of stack
ADD    R3, BYTE [25000]  ; Adds one byte of value at address 25000 to the register R3
SUB    R4, BYTE [0x0D0D]; Subtracts one byte of value at memory 0X0D0D from R4
INC    BYTE [12857]      ; Increments one byte of value at memory 12857
DEC    BYTE [13598]      ; Decrements one byte of value at memory 13598
JMP    42069             ; Jump to memory location 42069
```

Table-2 represent the page table of the process, Time required for **regular memory access(no page fault)** is 150ns, **servicing interrupt** is 2ms, **swapping out** is 10ms and **swapping in** is 5ms.

p	**f**	**d**	**r**	$v/\bar{i}$
0	000	0	0	0
1	010	0	0	0
2	000	1	0	1
3	010	0	1	1
4	000	0	1	0
5	111	0	0	0
6	110	0	1	1
7	001	1	0	1
8	011	1	0	0
9	110	0	0	0
10	000	0	1	0
11	100	0	0	0
12	100	0	0	1
13	000	1	0	0
14	111	0	0	0
15	101	1	0	1

Table 2: Page Table

(a) What will the assembly code look like after address binding? [4]

(b) **Find** the time required by the program to execute [4]

(c) What will the page table look like after the execution of the process? [2]

[Note: p = page number, f = frame number, d = modified/dirty, r = referenced, $v/\bar{i}$ **= valid/invalid]**

6. In a system, there are currently three processes of size 10MB, 1GB, and 10GB. The system has a physical memory of 512GB with frame size of 4GB.

(a) Which frame allocation method is more applicable in this scenario? **Explain** in three sentences. [2]

(b) **Mention** two potential downsides of the allocation strategy in context of the above scenario. [1]

(c) What kind of fragmentation will occur here? **Explain** in two sentences. [2]