United International University
Department of Computer Science and Engineering
CSE 4509/CSI 309: Operating System Concepts/Operating Systems
Final Examination: Fall 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 Consider a system with paging-based memory management, whose architecture allows for a 256 MB virtual address space for processes. The size of logical pages and physical frames is 2 KB. The system has 4 GB of physical RAM. The system allows a maximum of 256 processes to run concurrently. Assuming the OS uses hierarchical paging and each page table entry requires an additional 7 bits (beyond the frame number) to store various flags. Assume page table entries are rounded up to the nearest byte.

a) What is the size of the offset? [1]
b) How many Virtual pages and Physical frames are there? [2]
c) Find the size of a single page table. [2]
d) Design a multi level page table for the given virtual memory. [2]
e) calculate the maximum memory space required to store the page tables of all processes in the system. [3]

2 A computer system has a virtual memory system with a **16-bit virtual address** space and a **12-bit offset**. The **physical memory uses a 14-bit address** space. The system employs a **page size of 2^{12} bytes.** A process runs and generates the following sequence of virtual memory accesses, where each access is performed on one or two virtual addresses provided in binary format. [5]

The following is a list of assembly-like memory access codes:

1. LOAD R1, 0b1100101010011000
2. STORE R2, 0b0011110110111101
3. ADD R1, 0b1101010010100000, 0b1010011001000100
4. LOAD R3, 0b0001010001010111
5. SUB R4, 0b1100111101011110, 0b0111010011010101
6. MOV R5, 0b1110000111111111
7. STORE R6, 0b0111010100100100
8. LOAD R2, 0b1010110011110111
9. ADD R7, 0b0000110101101110, 0b1111000011110000
10. SUB R8, 0b1111111111111111, 0b0101010101010101

Use the **Least Recently Used (LRU)** page replacement policy to simulate the memory accesses and determine the number of page hits and page misses. Assume that the page table is initially empty.

3	Consider the following code:	

```
#define NUM_MUTEXES 3
pthread_mutex_t mutexes[NUM_MUTEXES];
void* thread_function(void* arg) {
    int thread_id = *((int*)arg);
    int next_mutex = thread_id % NUM_MUTEXES;
    printf("Thread %d is starting.\n", thread_id);

    pthread_mutex_lock(&mutexes[thread_id - 1]);
    sleep(5);

    pthread_mutex_lock(&mutexes[next_mutex]);fashion

    pthread_mutex_unlock(&mutexes[next_mutex]);
    pthread_mutex_unlock(&mutexes[thread_id - 1]);

    return NULL;
}
int main() {
    pthread_t threads[NUM_MUTEXES];
    int thread_ids[NUM_MUTEXES] = {1, 2, 3};
    for (int i = 0; i < NUM_MUTEXES; i++) {
        pthread_mutex_init(&mutexes[i], NULL);
    }
    for (int i = 0; i < NUM_MUTEXES; i++) {
        pthread_create(&threads[i], NULL, thread_function, &thread_ids[i]);
    }
    for (int i = 0; i < NUM_MUTEXES; i++) {
        pthread_join(threads[i], NULL);
    }

    for (int i = 0; i < NUM_MUTEXES; i++) {
        pthread_mutex_destroy(&mutexes[i]);
    }
    return 0;
}
```

Draw a resource allocation graph for the above code and comment if a deadlock can occur or not. [5]

4 Consider the Table-1 snapshot of a system: [5]

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

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.

5

a) Given the challenges of external fragmentation, which file allocation method would you recommend for a system frequently handling large, variable-sized files? Explain your choice with reasons. [3]

b) Let's assume the disk has 200 tracks, numbered from 0 to 199. **The head starts at track 50.**
Request queue: [15, 180, 90, 135, 45, 110, 70, 10, 175]
Find out which algorithm will perform better among **Shortest Seek Time First and Scan**? [7]

6

A hard disk drive has the following characteristics:

- Average seek time = 8 milliseconds
- Average rotational speed = 7200 RPM (revolutions per minute)
- Transfer rate = 100 MB/s
- Track size = 500 sectors
- Sector size = 512 bytes

A file system operation needs to read a sector that is located **200 tracks away from the current read/write head position**. Find the total access time for reading the sector. [5]