Department of Computer Science and Engineering
Midterm Examination Summer 2023
CSE 321: Operating Systems

A

Duration: 1 Hour 15 Minutes **Total Marks:** 25

Answer the following questions.
Figures in the right margin indicate marks.

---

1. CO1

a) Why do we need cooperating processes, and what are the two models of interprocess communication? Additionally, **mention** the strengths and weaknesses of each approach? **[3]**

b) Discuss one drawback faced by multi-programmed OS architecture and suggest possible way(s) to overcome it. **[2]**

c) What is the purpose of system calls? **Which** of the following instructions should be privileged? **[1+1]**
i. Set value of timer **ii.** Read the clock **iii.** Clear memory **iv.** Issue a trap instruction. **v.** Turn off interrupts **vi.** Modify entries in device-status table

d) Find the output of the following code snippet. Your output should exactly match with the original output. **[3]**

```
int main() {
    pid_t child_pid;
    int global_a = 90, b = 11;
    char message[] = "Hello, from the ";
    printf("Parent process started\n");
    child_pid = fork();
    if (child_pid == -1) {
        printf(“Fork Failed\n”);
    } else if (child_pid > 0) {
        wait(NULL);
        global_a += 97;
        printf("%sAddition: %d + %d = %d\n", message,
global_a, b, global_a);
        printf("%sSubtraction: %d - %d = %d\n", message, b,
global_a, b);
    } else {
        b *= 7;
        printf("Multiplication: %d * %d = %d\n", global_a,
b, global_a);
        printf("Division: %d / %d = %d\n", b, global_a, b);
    }
    return 0;
}
```

2.
CO2

Processes	Arrival Time	Burst Time	Priority
P1	0	4	2
P2	6	4	1
P3	7	6	6
P4	7	1	3
P5	8	7	4
P6	19	7	5

a) Draw a Gantt chart and illustrate the execution of the process using the **Round Robin** scheduling algorithm **(time quantum = 5 units)**. **Calculate** the **average waiting** and **turnaround time.** **[3+2]**

b) Apply Preemptive Priority scheduling algorithm. **Draw** the Gantt chart and **Calculate** the **average waiting** and **turnaround time.** **[2+2]**

c) Compare the results and **identify** the most suitable scheduling algorithm in this scenario. **[1]**

3.
CO3

a) Explain data parallelism with an example. **[1.5]**

b) You are developing a real-time embedded system for a safety-critical application, such as an advanced driver assistance system (ADAS) for autonomous vehicles. The system's primary goal is to process sensor data, make critical decisions, and take actions in real-time to ensure the safety of passengers and pedestrians. However, you may assume there's no limitation of computational resources on the project you are working on. **[1.5]**

Based on the scenario, **which** multi-threading model would you recommend for implementation? **Provide** necessary justification.

c) A system has processes to execute of which **30%** is serial. If the number of cores is decreased from **9** to **4**, **Explain** the change in the performance. **[2]**

Department of Computer Science and Engineering
Midterm Examination Summer 2023
CSE 321: Operating Systems

B

Duration: 1 Hour 15 Minutes **Total Marks:** 25

Answer the following questions.
Figures in the right margin indicate marks.

---

1. CO1

a) Define the term "**interrupt**" in the context of operating systems. **Mention** one advantages and one disadvantage of a multiprocessor system. **[3]**

b) A process from its creation till its completion will go through various states. To enter different states, the process requires the decision of different types of scheduler. **State** the name of different schedulers for different process states with justification. **[2]**

c) What is the purpose of dual mode operation? **Which** of the following instructions should be privileged? **[1+1]**
i. Access I/O device **ii.** Set value of timer **iii.** Read the date in the calendar.
iv. Clear memory **v.** Switch from user to kernel mode **vi.** Turn off interrupts.

d) Find the output of the following code snippet. **Your output should exactly match with the original output.** **[3]**

```
int main() {
    pid_t child_pid;
    int global_a = 68, b = 10;
    char message[] = "Hello, from the ";
    printf("Parent process started\n");
    child_pid = fork();
    if (child_pid == -1) {
        printf("Fork Failed\n");
    } else if (child_pid > 0) {
        wait(NULL);
        b *= 38;
        printf("Multiplication: %d * %d = %d\n", global_a,
b, global_a);
        printf("Division: %d / %d = %d\n", b, global_a, b);
    } else {
        global_a += 98;
        printf("%sAddition: %d + %d = %d\n", message,
global_a, b, global_a);
        printf("%sSubtraction: %d - %d = %d\n", message, b,
global_a, b);
    }
    return 0;
}
```

2.
CO2

Processes	Arrival Time	Burst Time	Priority
P1	0	4	2
P2	6	4	1
P3	7	6	6
P4	7	1	3
P5	8	7	4
P6	19	7	5

a) Draw a Gantt chart and illustrate the execution of the process using the **Round Robin** scheduling algorithm **(time quantum = 5 units)**. **Calculate** the **average waiting** and **turnaround time.** **[3+2]**

b) Apply Preemptive Priority scheduling algorithm. **Draw** the Gantt chart and **Calculate** the **average waiting** and **turnaround time.** **[2+2]**

c) Compare the results and **identify** the most suitable scheduling algorithm in this scenario. **[1]**

3.
CO3

a) Explain task parallelism with an example. **[1.5]**

b) You are developing a lightweight, user-level threading library for a resource-constrained embedded system. The embedded system has limited processing power and memory, and it does not provide native support for multithreading at the kernel level. The primary goal is to allow concurrent execution of multiple tasks while minimizing the overhead of managing threads. **[1.5]**

Based on the scenario, **which** multi-threading model would you recommend for implementation? **Provide** necessary justification.

c) A system has processes to execute of which **32%** is serial. If the number of cores is decreased from **8** to **2**, **Explain** the change in the performance. **[2]**

Department of Computer Science and Engineering
Final Examination Summer 2023
CSE 321: Operating Systems

A

Duration: 1 Hour 50 Minutes **Total Marks:** 40

Answer the following questions.
Figures in the right margin indicate marks.

---

1. CO4

a) Imagine a computer lab with multiple computers, and equipped with printers. Students use these computers for various tasks and may want to print documents. However, there are only a limited number of printers available. How do you solve this issue using semaphore? Your answer should have the steps associated in solving the given scenario. [3]

b) What is meant by busy waiting in the implementation of a critical section solution, and why is it considered problematic? [1.5]

c) For Peterson's problem below conditions will be applied.

- There are two processes: P1 and P2. P2 gets to execute first.
- Each Statement takes 4ms to execute.
- Context Switch will occur after 12ms.
- Critical section contains 4 statements.
- Remainder section contains 3 statements.
- For P1: i=0 and j=1
- For P2: i=1 and j=0
- turn=0
- flag[0] = FALSE, flag[1] = TRUE

The structure of process Pi in Peterson's solution:

```
do{
      flag[i] = true;
      turn = j;
      while(flag[j] == true && turn == 1){
            //busy wait
      }
      //critical section
      flag[i] = false;
      //remainder section
}while(true);
```

Complete the table below for processes P_1 and P_2 using **Peterson's solution.** [3.5]

Process 1: i = 0, j = 1	Process 2: i = 1, j = 0

2. **CO4** **a)** Consider the following snapshot of a system:

		Max					Allocation			
		A	B	C	D		A	B	C	D
P1		7	0	1	3		7	0	0	2
P2		2	7	5	0		2	1	0	0
P3		2	12	5	6		0	6	3	3
P4		1	6	5	6		0	2	1	2

	Available		
A	B	C	D
1	6	4	4

i. **Calculate** the Need Matrix. Is this system in a **safe state**? If yes, then find the safe sequence using Banker's Safety algorithm otherwise, provide the necessary explanation. [1+3]

ii. What happens if the process **P4** requests at this moment for **(0, 4, 2, 0)?** Whether Banker's algorithm grants the request or not? [5]

b) Suppose, in a workplace, we have a set of resource types, R = {R1, R2, R3, R4} and a set of processes, P = {P1, P2, P3, P4, P5}. **R1, R2, R3, and R4** have **3, 2, 3, and 2** instances respectively.

- P1 is holding 2 instances of R1
- P2 is holding 1 instance of R3
- P3 is holding 1 instance of R4
- P5 requests 2 instances of R3
- P4 is holding1 instance of R4
- P3 requests 1 instance of R2
- P2 requests 1 instance of R1
- P2 is holding 1 instance of R2
- P1 is requesting 1 instance of R4
- P3 is holding 1 instance of R3
- P4 is holding 1 instance of R2

Construct a resource allocation graph for the above scenario and **identify the cycle (if any) and decide** whether there is a deadlock or not. [4]

3. **CO5** **a)** Given variable size memory (dynamic) partitions of **8 MB, 29 MB, 35 MB, and 48 MB** (in order, top to bottom), **apply** best-fit and worst-fit algorithms to place processes with the space requirement of **5 MB, 15 MB, 10 MB, 5 MB, 10 MB, 20 MB, 25 MB, and 15 MB** (in order). Which algorithm makes the most effective use of memory? [4+1]

b) How is paging efficiently used in main memory to increase throughput? Justify with necessary examples. [3]

c) Compare the following systems in terms of performance: [3]

1. A system with a hit ratio of 93%, associative lookup time of 32ns, and memory access time of 72ns.
2. A system with a hit ratio of 62%, associative lookup time of 9ns, and memory access time of 133ns.

4. CO5 **a)** Suppose there are two processes **P1 (32 B)** and **P2 (40 B)** with a page size of **8 B**. The main memory size is **96 B.** The page table for P1 and P2 are given below:

Page#	Frame#
0	3
1	10
2	2
3	0

Page table: P1

Page#	Frame#
0	5
1	8
2	11
3	1
4	6

Page table: P2

Find the corresponding physical addresses of the following logical addresses: [3]

a) Address 011010 of P1
b) Address 111110 of P2
c) Address 011001 of P2

b) Consider a computer with a main memory that has 5 frames and page reference string of 0-7 page **[2, 2, 2, 6, 5, 5, 4, 2, 0, 0, 6, 1, 5, 5, 3, 0, 0, 2, 6, 5]**. The page reference string represents the order in which the pages are accessed by a program. **Apply LRU** & **OPT** algorithm to **simulate** the page replacement that occurs when the main memory can hold at most 5 pages at a time. **Record** the number of **page faults** and compare the result. **Mention** which algorithm performs better in this scenario. [4+1]

Department of Computer Science and Engineering

Final Examination Summer 2023

CSE 321: Operating Systems

B

Duration: 1 Hour 50 Minutes **Total Marks:** 40

Answer the following questions.
Figures in the right margin indicate marks.

---

1. **CO4**

a) Imagine a computer lab with multiple computers, and equipped with printers. Students use these computers for various tasks and may want to print documents. However, there are only a limited number of printers available. How do you solve this issue using semaphore? Your answer should have the steps associated in solving the given scenario. [3]

b) How do semaphores and mutexes differ in their implementation. [1.5]

c) For Peterson's problem below conditions will be applied.

- There are two processes: P1 and P2. P2 gets to execute first.
- Each Statement takes 4ms to execute.
- Context Switch will occur after 12ms.
- Critical section contains 4 statements.
- Remainder section contains 3 statements.
- For P1: i=0 and j=1
- For P2: i=1 and j=0
- turn=0
- flag[0] = FALSE, flag[1] = TRUE

The structure of process Pi in Peterson's solution:

```
do{
      flag[i] = true;
      turn = j;
      while(flag[j] == true && turn == 1){
            //busy wait
      }
      //critical section
      flag[i] = false;
      //remainder section
}while(true);
```

Complete the table below for processes P_1 and P_2 using **Peterson's solution.** [3.5]

Process 1: i = 0, j = 1	Process 2: i = 1, j = 0
↓	

2. **CO4**

a) Consider the following snapshot of a system:

		Allocation				Max				Available		
		A	**B**	**C**		**A**	**B**	**C**		**A**	**B**	**C**
P0		5	2	2		9	9	8		7	10	5
P1		3	2	0		9	9	10				
P2		5	3	3		6	8	5				
P3		3	0	0		6	7	9				
P4		4	2	1		5	7	6				

i. **Calculate** the Need Matrix. Is this system in a **safe state**? If yes, then find the safe sequence using Banker's Safety algorithm otherwise, provide the necessary explanation. [1+3]

ii. What happens if the process **P4** requests at this moment for **(0, 3, 1)?** Whether Banker's algorithm grants the request or not? [5]

b) Suppose, in a workplace, we have a set of resource types, R = {R1, R2, R3, R4} and a set of processes, P = {P1, P2, P3, P4, P5}. **R1, R2, R3, and R4** have **3, 1, 4, and 2** instances respectively.

- P1 is holding 2 instances of R1
- P2 is holding 1 instance of R3
- P3 is holding 1 instance of R4
- P5 requests 2 instances of R3
- P4 is holding 1 instance of R4
- P3 requests 1 instance of R2
- P2 requests 1 instance of R1
- P2 is holding 1 instance of R2
- P1 is requesting 1 instance of R4
- P3 is holding 1 instance of R3
- P4 is holding 1 instance of R3.

Construct a resource allocation graph for the above scenario and **identify the cycle (if any) and decide** whether there is a deadlock or not. [4]

3. **CO5**

a) Given variable size memory (dynamic) partitions of **10 MB, 16 MB, 45 MB, and 49 MB** (in order, top to bottom), **apply** best-fit and worst-fit algorithms to place processes with the space requirement of **5 MB, 15 MB, 10 MB, 5 MB, 10 MB, 20 MB, 25 MB, and 15 MB** (in order). Which algorithm makes the most effective use of memory? [4+1]

b) How cache memory can be used to design an efficient paging hardware. Your answer should have the required diagram with the necessary justification. [3]

c) Compare the following systems in terms of performance: [3]

1. A system with a hit ratio of 72%, associative lookup time of 24ns, and memory access time of 56ns.
2. A system with a hit ratio of 65%, associative lookup time of 6ns, and memory access time of 133ns.

4. CO5 **a)** Suppose there are two processes **P1 (32 B)** and **P2 (40 B)** with a page size of **8 B**. The main memory size is **96 B.** The page table for P1 and P2 are given below:

Page#	Frame#
0	3
1	10
2	2
3	0

Page table: P1

Page#	Frame#
0	5
1	8
2	11
3	1
4	6

Page table: P2

Find the corresponding physical addresses of the following logical addresses: [3]

a) Address 000001 of P1
b) Address 001001 of P2
c) Address 101001 of P2

b) Consider a computer with a main memory that has 5 frames and page reference string of 0-7 page **[6, 5, 3, 6, 4, 5, 0, 5, 5, 1, 1, 5, 0, 5, 4, 6, 3, 5, 1, 5]**. The page reference string represents the order in which the pages are accessed by a program. **Apply LRU** & **OPT** algorithm to **simulate** the page replacement that occurs when the main memory can hold at most 5 pages at a time. **Record** the number of **page faults** and compare the result. **Mention** which algorithm performs better in this scenario. [4+1]