Department of Computer Science and Engineering
Midterm Examination Spring 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) **Explain** why main memories are usually volatile. [2]

b) **Distinguish** between monolithic and microkernel OS structures. [3]

c) A program has a process that will allow its child process to complete first. **Which** system call can be used in this scenario? **Explain** what may happen in absence of this system call. [2]

d) **Find** the output of the following code snippet. [3]

```
const int len = 2;
int main(){
      int id;
      int a[] = {10,7};
      int b = len-1;
      id = fork();
      if (id < 0){
            printf("fork failed\n");
      }
      else if(id == 0){
            printf("child process executing\n");
      }
      else{
            wait(NULL);
            printf("parent process executing\n");
            a[b-1]=a[b-1]-2;
            a[b] = a[b]+2;
      }
      for(int i=0;i<len;i++){
            printf("value of a[%d]: %d\n",i,a[i]);
      }
      return 0;
}
```

2.
CO2

Processes	Arrival Time	Burst Time
P1	0	11
P2	20	6
P3	14	9
P4	20	8
P5	15	8
P6	16	8
P7	2	2

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 Shortest Remaining Time First (SRTF)** 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) A system has processes to execute of which **35%** is parallel. If the number of cores is increased from **3 to 5**, Explain what will be the increase/decrease in performance? [2]

b) A program has multiple threads that need to perform a task which involves similar computation on a large scale of data. **Identify** which kind of parallelism can be applied in this scenario. **Provide** proper justification with a **real-life example** of the scenario mentioned above. [3]

Department of Computer Science and Engineering
Midterm Examination Spring 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) **Briefly explain** dual mode operation of OS. [2]

b) **Distinguish** between layered and microkernel OS structures. [3]

c) Two processes (A and B) are distributed across multiple physical machines or networked systems. **Which** technique can be used to achieve inter-process communication in this scenario? Is it suitable for exchanging large amounts of data? **Provide** proper justification to support your answer. [2]

d) **Find** the output of the following code snippet. [3]

```
const int len = 2;
int main(){
      int id;
      int a[] = {5,8};
      int b = len-1;
      id = fork();
      if (id < 0){
            printf("fork failed\n");
      }
      else if(id > 0){
            wait(NULL);
            printf("parent process executing\n");
      }
      else{
            printf("child process executing\n");
            a[b-1]=a[b-1]+2;
            a[b] = a[b]-3;
      }
      for(int i=0;i<len;i++){
            printf("value of a[%d]: %d\n",i,a[i]);
      }
      return 0;
}
```

2.
CO2

Processes	Arrival Time	Burst Time
P1	0	11
P2	20	6
P3	14	9
P4	20	8
P5	15	8
P6	16	8
P7	2	2

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]

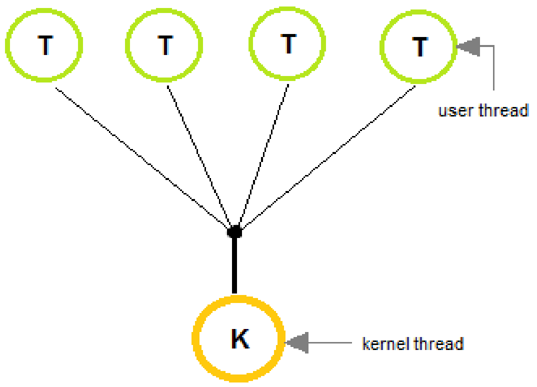
b) **Apply Shortest Remaining Time First (SRTF)** scheduling algorithm. **Draw** the Gantt chart and **Calculate** the **average waiting** and **turnaround time.** [2+2]

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

3.
CO3

a) A system has processes to execute of which **45%** is parallel. If the number of cores is increased from **2** to **4**, **Explain** what will be the increase/decrease in performance. [2]

b) **Describe** the multithreading model which is shown in the picture below. **Identify** the issue in this model and **suggest** another multithreading model which is free from that issue. [3]



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

A

Duration: 2 Hours **Total Marks:** 40

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

1.
CO4

a) In a restaurant, there are 3 washrooms available for male guests and 1 washroom available for female guests. There is a strict rule maintained by the authorities that neither men can use the female washroom nor women can use any of the male washrooms. On a random day during lunch time, 9 male guests needed to use washroom facilities at the very same time. But none of them were allowed to use the female washroom although it was vacant. Therefore, only 3 guests were able to get access to washrooms at a time and others had to wait while maintaining a queue. If any of the washrooms gets vacant, a person from the queue can get access to that. Logically **explain** which synchronization method has been used here. [3]

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

- There are two processes: P_1 and P_2.
- Each Statement takes 4ms to execute, P_1 gets executed first
- Context Switch will occur after 16ms.
- Critical section contains 4 statements.
- Remainder section contains 2 statements.
- For P_1: i = 0 and j = 1
- For P_2: 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 given below for processes P_1 and P_2 using **Peterson's solution.** [4]

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

2. **CO4**

a) Consider the following snapshot of a system:

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

Available		
A	**B**	**C**
8	5	6

i. Is the system in a safe state? **Apply Banker's safety algorithm** to find out the safe sequence. You need to calculate the need matrix. [4]

ii. **P_0** requests for **(1 4 1), check** the validity of the request. If the request is valid, does the system enter a **deadlock**? [1+4]

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 **2, 3, 1, and 2** instances respectively.

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

Construct a resource allocation graph for the above scenario. **Mention** the number of cycles found and **identify** whether there is a deadlock or not. [4]

3. **CO5**

a) Given fixed size memory partitions of 300k, 480k, 110k, 200k, 360k, and 550k **(in order, bottom to top**), **apply** first-fit and best-fit algorithms to place processes with the space requirement of 426k, 300k, 125k, 104k, 475 and 340k (in order). Which algorithm makes the most effective use of memory? Is there any external fragmentation? [4+1+1]

b) Considering in Dynamic memory management technique at a certain time the memory looks like the following figure: [2]

OS	P3 = 200K	200K	P6 = 600K	200K	P7 = 200K

Where gray portions of the memory are free spaces. A process P5 = 300K arrives in the ready queue. How can you accommodate P5 in the memory? **Justify** your answer.

c) If the page size is **3 KB**, **how many** frames will be needed in Main memory for a process size of **32,167 Bytes**? Is there any **internal fragmentation**? - If yes, **calculate** the value. [1 KB = 1024 Bytes] [2]

4. CO5 **a)** Suppose there are two processes **P1 (16 B)** and **P2 (12 B)** with a page size of **4 B**. The main memory size is **40 B.** The page table for P1 and P2 are given below:

Page#	Frame#	Contents
0	7	CSE110
1	4	CSE111
2	6	CSE221
3	5	CSE321

Page table: P1

Page#	Frame#	Contents
0	8	CSE110
1	0	CSE111
2	3	CSE220

Page table: P2

i. **Draw** the memory representations consisting of the contents for both P1 and P2. [2]

ii. **Find** the corresponding physical addresses of the following logical addresses: [2]
   a) Address 1010 of P1
   b) Address 0111 of P2

iii. **How** can you make efficient use of the main memory in this scenario? [1]

b) Consider a computer with a main memory that has 4 frames and page reference string of 0-7 pages: **[2, 5, 7, 5, 2, 0, 2, 5, 6, 2, 6, 6]**. The page reference string represents the order in which the pages are accessed by a program. **Apply FIFO** & **OPT** algorithm to **simulate** the page replacement that occurs when the main memory can hold at most 4 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 Spring 2023

CSE 321: Operating Systems

B

Duration: 2 Hours | **Total Marks:** 40

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

---

1. CO4

a) In a restaurant, there are 3 washrooms available for male guests and 1 washroom available for female guests. There is a strict rule maintained by the authorities that neither men can use the female washroom nor women can use any of the male washrooms. On a random day during lunchtime, 4 female guests needed to use washroom facilities at the very same time. But none of them were allowed to use any of the male washrooms although two of them were vacant. Therefore, only 1 guest was able to get access to the washroom at a time and others had to wait while maintaining a queue. If the washroom gets vacant, a guest from the queue can get access to that. Logically **explain** which synchronization method has been used here. [3]

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

- There are two processes: P_1 and P_2.
- Each Statement takes 5ms to execute, P_1 gets executed first
- Context Switch will occur after 20ms.
- Critical section contains 2 statements.
- Remainder section contains 4 statements.
- For P_1: i = 0 and j = 1
- For P_2: 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 given below for processes P_1 and P_2 using **Peterson's solution.** [4]

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

2. **CO4**

a) Consider the following snapshot of a system:

		Allocation				MAX		
		A	**B**	**C**		**A**	**B**	**C**
P0		5	2	3		9	9	8
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

Available		
A	**B**	**C**
7	10	5

i. Is the system in a safe state? **Apply Banker's safety algorithm** to find out the safe sequence. You need to calculate the need matrix. [4]

ii. P_4 requests for **(0 3 1)**, **check** the validity of the request. If the request is valid, does the system enter a **deadlock**? [1+4]

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 **2, 3, 1, and 2** instances respectively.

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

Construct a resource allocation graph for the above scenario. **Mention** the number of cycles found and **identify** whether there is a deadlock or not. [4]

3. **CO5** **a)** Given fixed size memory partitions of 210k, 350k,250k,190k, 250k and 452k **(in order, bottom to top)**, **apply** first-fit and best-fit algorithms to place processes with the space requirement of 250k, 425k, 212k, 160k, 210 and 440k (in order). **Which** algorithm makes the most effective use of memory? **Is there** any external fragmentation? [4+1+1]

b) Considering in Dynamic memory management technique at a certain time the memory looks like the following figure: [2]

OS	P3 = 150K	150K	P6 = 300K	150K	P7 = 150K

Where gray portions of the memory are free spaces. A process P5 = 150K arrives in the ready queue. How can you accommodate P5 in the memory? **Justify** your answer.

c) If the page size is **3 KB**, **how many** frames will be needed in Main memory for a process size of **79,071 Bytes**? Is there any **internal fragmentation**? - If yes, **calculate** the value. [1 KB = 1024 Bytes] [2]

4. **CO5** **a)** Suppose there are two processes **P1 (16 B)** and **P2 (12 B)** with a page size of **4 B**. The main memory size is **40 B.** The page table for P1 and P2 are given below:

Page#	Frame#	Contents
0	9	UB2
1	0	UB8
2	2	UB1
3	7	UB7

Page table: P1

Page#	Frame#	Contents
0	4	UB2
1	3	UB1
2	6	UB4

Page table: P2

i. **Draw** the memory representations consisting of the contents for both P1 and P2. [2]

ii. **Find** the corresponding physical addresses of the following logical addresses: [2]
   a) Address 1010 of P1
   b) Address 0111 of P2

iii. **How** can you make efficient use of the main memory in this scenario? [1]

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