Student ID & Name:

____

Zhejiang University of Technology

Realtime Systems and Operating Systems

Examiners: Fuli Wu

Time allowed: 1.5 hours

Instructions:
There is a total of 50 marks available.

	Section A	Section B	Section C	Total
Score				

SECTION A

Attempt all questions. All questions in this section are worth 18 marks.

Question 1 *[3 marks]*
What are the three main purposes of an operating system?

Question 2 *[5 marks]*
Of the following five forms of storage, rank them from fastest to slowest in terms of access time: (1) main memory, (2) magnetic disk, (3) registers, (4) solid state disk, (5) cache.

Question 3 *[5 marks]*
List at least five different types of computing environments.

Question 4 *[5 marks]*
As a process executes, it changes state. The state of a process is defined in part by the current activity of that process. A process may be in one of five states. Please list the five states and illustrate the diagram of process state.

SECTION B

Attempt all questions. All questions in this section are worth 16 marks.

Question 1 *[2 marks]*
Consider a logical address space of 1024 pages of 1024 bytes each, mapped onto a physical memory of 512 frames.

a. How many bits are there in the logical address?
b. How many bits are there in the physical address?

Question 2 *[2 marks]*
Illustrate how a binary semaphore can be used to implement mutual exclusion among n processes.

Question 3 *[2 marks]*

Including the initial parent process, how many processes are created by the program

```
#include <stdio.h>
#include <unistd.h>
int main()
{
    int i;
    for (i = 0; i < 6; i++)
        fork();
    return 0;
 }
```

Question 4 *[4 marks]*

Using the program shown in Figure 3.35, explain what the output will be at lines X and Y.

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

#define SIZE 6

int nums[SIZE] = {0, 2, 4, 6, 8, 10};

int main()
{
    int i;
    pid t pid;
    pid = fork();
    if (pid == 0) {
        for (i = 0; i < SIZE; i++) {
            nums[i] *= -i;
            printf("CHILD: %d ",nums[i]); /* LINE X */
        }
    }
```

```
        else if (pid > 0) {
            wait(NULL);
            for (i = 0; i < SIZE; i++)
                printf("PARENT: %d ",nums[i]); /* LINE Y */
        }
        return 0;
    }
```

Question 5 *[6 marks]*
Suppose that the following processes arrive for execution at the times indicated. Each process will run for the amount of time listed. In answering the questions, use non preemptive scheduling, and base all decisions on the information you have at the time the decision must be made.

Process	Arrival Time	Burst Time
P1	0.0	8
P2	0.4	5
P3	1.0	2

1. What is the average turnaround time for these processes with the FCFS scheduling algorithm?
2. What is the average turnaround time for these processes with the SJF scheduling algorithm?
3. The SJF algorithm is supposed to improve performance, but notice that we chose to run process P1 at time 0 because we did not know that two shorter processes would arrive soon. Compute what the average turnaround time will be if the CPU is left idle for the first 1 unit and then SJF scheduling is used. Remember that processes P1 and P2 are waiting during this idle time, so their waiting time may increase. This algorithm could be known as future-knowledge scheduling.

SECTION C

Attempt all questions. All questions in this section are worth 16 marks.

Question 1 *[8 marks]*

The following processes are being scheduled using a preemptive, round- robin scheduling algorithm. Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes listed below, the system also has an idle task (which consumes no CPU resources and is identified as Pidle). This task has priority 0 and is scheduled whenever the system has no other available processes to run. The length of a time quantum is 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue.

Process	Arrival Time	Burst	Priority
P1	0	20	40
P2	25	25	30
P3	30	25	30
P4	65	15	35
P5	100	10	5
P6	105	10	10

a. Show the scheduling order of the processes using a Gantt chart.
b. What is the turnaround time for each process?
c. What is the waiting time for each process?
d. What is the CPU utilization rate?

Question 2 *[8 marks]*

Suppose that a database is to be shared among several concurrent processes. Some of these processes may want only to read the database, whereas others may want to update (that is, to read and write) the database. We distinguish between these two types of processes by referring to the former as readers and to the latter as writers. To ensure that these difficulties do not arise, we require that reading and writing can't access the shared database simultaneously. This synchronization problem is referred to as the readers – writers problem.

The simplest readers–writers problem, requires that no reader be kept waiting unless a writer has already obtained permission to use the shared object. Please present a solution to the first readers–writers problem. In the solution, the reader and writer processes share the following data structures:

```
semaphore rw_mutex = 1;
semaphore mutex = 1;
int read_count = 0;
```

Please write the pseudo-code for the structure of a writer process and the structure of a reader process.