操作系统作业 2

要求：独立完成，严禁抄袭，按时提交。

提交截至日期：5.11（周日）18:00。

提交方式：提交 PDF 版本，PDF 命名规则为学号_姓名_作业 2，比如 PB200110000_张三_作业 2.pdf。

1. Explain why “terminated state” is necessary for processes. Explain what a zombie process is and how to eliminate a zombie process (i.e., remove its PCB entry from kernel).

2. What are the benefits of multi-threading? Which of the following components of program state are shared across threads in a multithreaded process?
   a. Register values
   b. Heap memory
   c. Global variables
   d. Stack memory

3. Consider the following code segment:

```
pid t pid;
pid = fork();
if (pid == 0) { /* child process */
        fork();
        thread create( . . .);
}
fork();
```

a. How many unique processes are created?

b. How many unique threads are created?

4. The program shown in the following figure uses Pthreads. What would be the output from the program at `LINE C` and `LINE P`?

```
#include <pthread.h>
#include <stdio.h>

int value = 0;
void *runner(void *param); /* the thread */

int main(int argc, char *argv[])
{
pid_t pid;
pthread_t tid;
pthread_attr_t attr;

   pid = fork();

   if (pid == 0) { /* child process */
      pthread_attr_init(&attr);
      pthread_create(&tid,&attr,runner,NULL);
      pthread_join(tid,NULL);
      printf("CHILD: value = %d",value); /* LINE C */
   }
   else if (pid > 0) { /* parent process */
      wait(NULL);
      printf("PARENT: value = %d",value); /* LINE P */
   }
}

void *runner(void *param) {
   value = 5;
   pthread_exit(0);
}
```

5. What are the two abstract models of IPC? Explain their pros and cons.

6. What are the differences between ordinary pipe and named pipe?

7. What is race condition and how to address it?

8. List all the three requirements of the entry and exit implementation when solving the critical-section problem.

9. For Peterson's solution, prove that it satisfies the three requirements of a solution for the critical section problem. Analyze whether strict alternation satisfies all the requirements.

10. What is deadlock? List the four requirements of deadlock.

11.Consider the following snapshot of a system:

	Allocation	*Max*
	A B C D	*A B C D*
T_0	1 2 0 2	4 3 1 6
T_1	0 1 1 2	2 4 2 4
T_2	1 2 4 0	3 6 5 1
T_3	1 2 0 1	2 6 2 3
T_4	1 0 0 1	3 1 1 2

Using the banker's algorithm, determine whether or not each of the following states is unsafe. If the state is safe, illustrate the order in which the threads may complete. Otherwise, illustrate why the state is unsafe.

a. Available = (2, 2, 2, 3)

b. Available = (4, 4, 1, 1)

c. Available = (3, 0, 1, 4)

d. Available = (1, 5, 2, 2)

12.What is semaphore? Explain the functionalities of semaphores. Please use semaphore to provide a deadlock-free solution to address the dining philosopher problem.

13. Consider the following set of processes, with the length of the CPU burst time given in milliseconds:

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.

a) Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF (nonpreemptive), nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1).

b) What is the turnaround time of each process for each of the scheduling algorithms in part a)?

c) What is the waiting time of each process for each of these scheduling algorithms?

d) Which of the algorithms results in the minimum average waiting time (over all processes)?

e) Illustrate the pros and cons of the algorithms: FCFS, SJF, priority scheduling and RR.

14. Illustarte the key ideas of rate-monotonic scheduling and earliest-deadline-first scheduling. Give an example to illustrate under what circumstances rate-monotonic scheduling is inferior to earliest-deadline-first scheduling in meeting the deadlines associated with processes?