1. (6 points) Explain what are preemptive scheduling and non-preemptive scheduling? What are design tradeoffs between preemptive scheduling and non-preemptive scheduling?

2. (8 points) Suppose that processes with the following CPU bursts have arrived on this order in the ready queue at time 0: A(4), B(3), C(5), D(7), E(1), F(9); There is no other processes on this system with a single core.

Draw a CPU Gantt chart and calculate the average waiting times for each of the following scheduling algorithms.

Show your waiting time calculation.
   a. FCFS (First Come First Served)
   b. SJF (Shortest Job First, non-preemptive)
   c. RR (Round Robin) with a quantum of 2, assuming no switching overhead

3. (8 points) Suppose that the following processes with given CPU bursts are available at time 0: A(6), B(5), C(4), D(2), E(9), F(2); For a system with a dual-core processor, draw the CPU Gantt chart of the schedules for the following scheduling algorithms:
   a. FCFS (First Come First Served)
   b. SJF (Shortest Job First)

Moreover, what is the minimal schedule length you may get and what is the corresponding order of the tasks being executed?

4. (6 points) What are possible problems in the following program if we are designed it for running forever? How can you solve it (show the fixed program)? Assuming fork() is always successful here.

```c
#include <signal.h>
#include <sys/wait.h>
int main() {
```
for (; ;) {
    if (!fork() ) { exit(0); }
}

5. (12 points) Writing two programs, producer.c and consumer.c, and these two programs are using the shared memory to communicate. Producer is writing a string “Hihi, Producer” to the same memory segment. Consumer is going to read from this memory segment and print the result. In the end of consumer, you should remove the created shared memory segment in the system in you code. You can confirm this by using “ipcs –m”.

Tips: To create a memory segment connecting with two processes, you may use the following API:
key_t ftok(const char *pathname, int proj_id);

Note: Provide the code in your handout. You can remove all those header files to save space. I suggest you to actually run this program on the system.