1. **Questions with short answers.** (6×5 points)
   
   a) What is the major advantage of microkernels?
   
   Microkernels allow kernel extensions to run in user space, which prevents them from crashing the kernel. *(Also: Microkernels are smaller, more manageable and easier to secure.)*
   
   b) What is the main function of timer interrupts?
   
   Timer interrupts prevent processes from monopolizing the CPU for long periods of time.
   
   c) What is the main disadvantage of non-preemptive scheduling policies?
   
   Processes that do not do issue system calls can monopolize the CPU for long periods of time.
   
   d) What is the main advantage of the symmetric organization for multiprocessor operating systems?
   
   All processors can run kernel code, which avoids one potential bottleneck.
   
   e) What is the main disadvantage of delaying disk writes?
   
   Writes may be lost if the process—or the system—crashes.
   
   f) Give an example of a hard real-time application?
   
   Heart pacemakers, and industrial process control systems.

2. What is the **default action** a process takes when it **receives a signal**? (5 points) How can processes specify which other action to take? (5 points) Is it always possible? (5 points)
   
   a) By default, a process that receives a signal terminates.
   
   b) A process can specify any other action to be taken when it receives a signal by executing beforehand a signal() system call. The signal is then said to be caught by the process.
   
   c) The ninth SIGKIL cannot be caught and always forces receiving processes to terminate.

3. Which are the three required conditions to prevent unauthorized access to user data in a shared computer system? (3×5 points) *(Hint: these requirements can be hardware, software or anything else.)*
   
   a) The system must have a dual-mode CPU in order to prevent user processes from directly accessing the disk and other peripherals.
   
   b) The system must have memory protection in order to prevent malevolent users from tampering with the kernel code (and the code of other running processes).
   
   c) Users must be prevented from rebooting the system with a rogue operating system.
4. Complete the following scheduling table for a System V Release 4 scheduler using the most reasonable values for all parameters: (5 points per correct line)

<table>
<thead>
<tr>
<th>#ts_quantum</th>
<th>ts_tqexp</th>
<th>ts_slpret</th>
<th>ts_maxwait</th>
<th>ts_lwait</th>
<th>LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>0</td>
<td>1</td>
<td>4000</td>
<td>1</td>
<td># 0</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>1</td>
<td>1000</td>
<td>1</td>
<td># 1</td>
</tr>
</tbody>
</table>

5. How many lines will the following program print? (5 points)

```c
#include <stdio.h>
main() {
    int pid;
    fork();
    fork();
    if ((pid = fork()) == 0)
        printf("Hello!\n");
} // main
```

Answer: The program will print **four** lines.

6. Complete the following code fragment in order to have stdout redirected to the pipe `piped`? (2×5 points)

```c
int piped[2];
pipe(piped);
close( _____ 1 _____ );
dup( _____ piped[1] _____ );
```

7. Which are the three states from where a process can enter the ready queue? (3×5 points)
   a) From the **running** state when the process gets interrupted by the scheduler
   b) From the **waiting** state after the process completes a system call
   c) From the **new** state after the process arrives