長庚大學107學年度第二學期 作業系統實務 期末測驗（總分104）

<<請依題號順序作答，跳號作答不予計分>>

系級: 姓名: 學號:

1. (10%) Assume that there are a public key K_e and a private key K_d, where K_e and K_d are a pair. E() and D() are the encryption and decryption functions, respectively. E(K_e, X) is the encryption result of any data X by using function E() with the key K_e, and D(K_d, Y) is the decryption result of any ciphertext Y by using function D() with the key K_d. Now, let Emilly have key K_e, and David have key K_d.
   (a) If Emilly wants to send private data Q to David, what should they do?
   (b) If David wants to prove that he is David (i.e., having the private key K_d) to Emilly, what should they do?

Answer: (a) 1. Emilly sends the encryption result E(K_e, Q) to David.
2. David then gets the decryption result D(K_d, E(K_e, Q)) → Q.
(b) 1. Emilly sends the encryption result E(Ke, P) to David.
2. David then gets the decryption result D(Kd, E(Ke, P)) → P.
3. David sends the result P back to Emilly to prove that David has the private key.

2. (10%) In lab exercises, we have compiled the Linux kernel, prepared the NFS root filesystem, and booted the evaluation board with the kernel and filesystem. To do so, we need to give some commands to the host system (PC) and to the evaluation board (實驗板). For the following list of commands, please answer that (a) which commands should be conducted on the host system, and (b) which commands should be conducted on the evaluation board with minicom.
   (1) *export PATH=$PATH:/opt/usr/local/arm/3.3.2/bin*
   (2) *make uImage*
   (3) *set bootargs console=ttyS0,115200n8 rw ip=192.168.68.yy root=/dev/nfs nfsroot=192.168.68.zz:/tmp/rootfs2.6,v3*
   (4) *tftpboot 0x10000000 uImage*
   (5) *bootm 0x10000000*
   (6) *yum –y install tftp-server tftp*
   (7) *yum -y install nfs-utils*
   (8) *exportfs -rv*
   (9) *insmod testmod.ko*
   (10) *saveenv*

Answer: (a) 1, 2, 6, 7, 8
(b) 3, 4, 5, 9, 10

3. (10%) To use the LEDs and Timer on the evaluation board we need the following information:
   - Include the header file: <asm-arm/arch-omap/tps65010.h> and <linux/timer.h>
   - Use the function: tps65010_set_led(which_LED , what_operation)
   - The LEDs on the board: LED1 and LED2
   - The operations of an LED: OFF, ON and BLINK
   - ***jiffies*** is the counter for the timer interrupts
   - ***HZ*** is the number of timer interrupts per second

   Now, we have the following source code of a Linux kernel module:

```
#include <linux/init.h>
#include <linux/module.h>
MODULE_LICENSE("License for you");

static int mymodule_init(void) {
        printk("Instert My Module to the Linux Kernel!\n");
        return 0;
}
static void mymodule_exit(void) {
        printk("My Module is Unloaded!\n");
}
module_init(mymodule_init);
module_exit(mymodule_exit);
```

Please modify the kernel module to meet the following requirement:

- Make LED 1 “BLINK” at the beginning of the insertion
- The duration of BLINK is 5 seconds
- After the insertion, make LED 1 “ON”
- Do not modify the process of module deletion

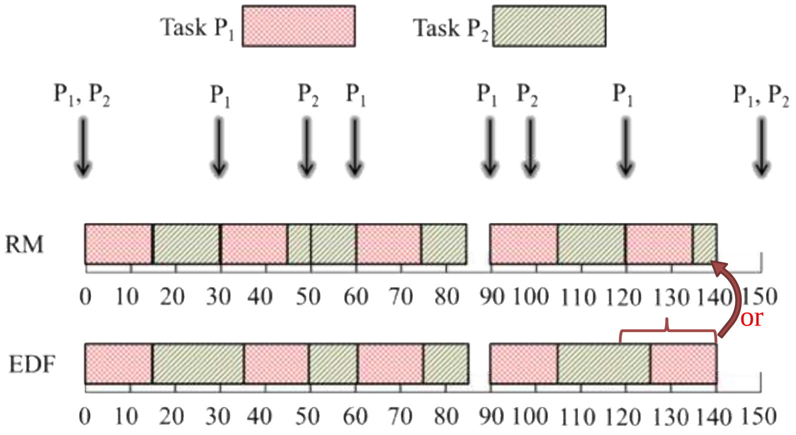
Answer:

```
#include <linux/init.h>
#include <linux/module.h>
#include <asm-arm/arch-omap/tps65010.h>
#include <linux/timer.h>
MODULE_LICENSE("License for you");
int a;
static int mymodule_init(void)
{
        tps65010_set_led(LED1 , BLINK);
        a = jiffies;
        while(a + 5*HZ > jiffies) ;
        tps65010_set_led(LED1, ON);
        return 0;
}
static void mymodule_exit(void)
{
        printk("My Module is Unloaded!\n");
}
module_init(mymodule_init);
module_exit(mymodule_exit);
```

4. (12%) Please briefly explain (a) the **Rate Monotonic (RM)** scheduling algorithm and (b) the **Earliest Deadline First (EDF)** scheduling algorithm for real-time task scheduling. For two periodic tasks P_1 and P_2, P_1 has its period 30 and execution time 15, and P_2 has its period 50 and execution time 20. Assume P_1 and P_2 are ready at time 0. Please draw the scheduling results from time 0 to 150 for (c) the RM scheduling and (d) the EDF scheduling.

Answer: (a) A static priority is assigned to each task based on the inverse of its period. (3%)
(b) Dynamic priorities are assigned according to deadlines. A task with the earliest deadline is assigned the highest priority. (3%)

(c) (3%) (d) (3%)



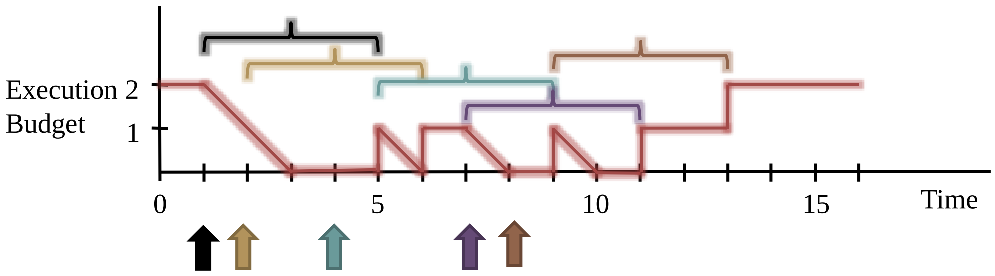
5. (10%) Consider 4 tasks, t_1, t_2, t_3, and t_4 which have priorities x_1, x_2, x_3, and x_4, respectively, and assume $x_1>x_2>x_3>x_4$ (x_1 is the highest priority). After we profiled the programs of the 4 tasks, we have the following information:
    - Task t_1 will lock semaphore S_1 for 40 ms.
    - Task t_2 will lock semaphore S_1 for 20 ms and lock semaphore S_2 for 50 ms.
    - Task t_3 will lock semaphore S_1 for 10ms and lock semaphore S_3 for 60ms.
    - Task t_4 will lock semaphore S_2 for 30ms and lock semaphore S_4 for 70ms.

    (a) Please derive the priority ceiling of each semaphore. If the priority ceiling protocol is used to manage the semaphore locking, (b) please derive the worst-case blocking time of each task. You have to provide the reason to support each of your answers.

Answer: (a) S_1: x_1, S_2: x_2, S_3: x_3, S_4: x_4.. You have to provide the season to support your answers.
(b) t_1: 20ms, t_2: 30ms, t_3: 30ms, t_1: 0ms.You have to provide the season to support your answers.

6. (12%) A sporadic server has a replenishment period 4 and an execution budget 2. Let the sporadic server have the budget 2 at time 0. Assume that events arrive at 1, 2, 4, 7, 8, and each event consumes the execution time 1. Please draw a diagram to show the changing of the execution budget at different time points.

Answer:



7. (10%) When the OS **µC/OS-II** is running, (a) can we directly install a new application on it? (b) If your answer is yes, please explain the procedure for installing a new application. If your answer is no, please describe how can we have a new application on **µC/OS-II**.

Answer: (a) No (4%)
(b) We have to collect the source files of µC/OS-II and all applications (including the new application) and compile them to have the new image. The system should be turned off, and the image is then install on the system. After that, we can boot the system with the new application running. (6%)

8. (10%)What is the “proc” filesystem of Linux?

Answer: The proc filesystem is a special filesystem in Unix-like operating systems that presents information about processes and other system information in a hierarchical file-like structure, providing a more convenient and standardized method for dynamically accessing process data held in the kernel than traditional tracing methods or direct access to kernel memory. It can be used to collect some hardware and system information via some user file I/O requests.

9. (10%) For real-time scheduling with shared resources, we need to further use semaphores for protecting the shared resources. Thus, during the execution of a task, it might be blocked or preempted. Please define (a) blocking and (b) preemption in real-time scheduling with synchronization.

Answer: (a) A higher-priority process is forced to wait for the execution of a lower-priority process because of some resource usage. The rules are defined by some resource management protocol.
(b) A low-priority process is forced to wait for the execution of a high-priority process. The priorities are assigned by some scheduling algorithm.

10. (10%) Worst-Case Execution Time (WCET) analysis is very important and essential for real-time system designs. In order to provide access information to data-cache and pipeline, detect infeasible paths, and derive loop bounds, we need to conduct Value Analysis. Suppose the abstract Interpretation of valuables x and y are [-3, 4] and [2, 5], respectively. Please specify the ranges of valuables j and k, where $j=x+y$ and $k=y-x$.

Answer: j = a+b = [-3, 4] + [2, 5] = [-1, 9]
k = b-a [2, 5] – [-3, 4] = [-2, 8]