# The University of Nottingham Ningbo China

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, SPRING SEMESTER 2015-2016

**AE2OSC, Operating Systems and Concurrency**

Time allowed: TWO hours

---

*Candidates may complete the front cover of their answer book and sign their desk card but must NOT write anything else until the start of the examination period is announced*

**You must answer THREE questions out of FOUR**
**(Only the THREE nominated solutions will be marked)**

*Only silent, self-contained calculators with a Single-Line Display are permitted in this examination.*

*Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject specific translation dictionaries are not permitted.*

*No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.*

***DO NOT turn your examination paper over until instructed to do so***

**ADDITIONAL MATERIAL:** None

**INFORMATION FOR INVIGILATORS:**
Please collect the exam papers at the end of the exam.

**Question 1 (Processes)**

**[25 marks]**

a) List the 5 process states and explain all the transitions between them.

[5 marks]

b) Briefly explain what a context switch is (i.e. **not** a mode switch) and describe the different actions that you would expect an operating system to take to carry out a context switch.

[4 marks]

c) List two examples where it is important for the operating system to take the architecture of the hardware into account.

[3 marks]

d) A multi-threaded CPU bound process is running on a machine with a multi-core CPU. Ignoring the overhead of context switching and thread creation, would you expect the process to finish quicker when user level threads are used, or when kernel level threads are used. Briefly explain your answer.

[3 marks]

e) The code below is supposed to create as many processes as specified by `NUMBER_OF_PROCESSES`. Explain how many processes it actually creates and how you would modify the code to get the exact number of processes as defined by `NUMBER_OF_PROCESSES` (if you would modify it at all).

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

#define NUMBER_OF_PROCESSES 2

int main() {
        pid_t pid = 0;
        int i;
        for(i = 0; i < NUMBER_OF_PROCESSES; i++)
        {
                pid = fork();
                if(pid < 0) {
                        printf("Could not create process\n");
                        exit(1);
                } else if(pid == 0) {
                        printf("Hello from the child process\n");
                }
        }
}
```

[4 marks]



continued on next page

f) The following questions are about the *Shortest Job First* algorithm and a *Priority Queue algorithm* for the processes given below (assume all processes arrive at the same time).

| | FCFS Position | CPU burst time (ms) | Priority |
|---|---|---|---|
| Process A | 1 | 60 | 2 (low) |
| Process B | 2 | 35 | 2 (low) |
| Process C | 3 | 15 | 1 (high) |
| Process D | 4 | 20 | 1 (high) |

i. Illustrate times when processes start and finish running respectively (you can use Gantt charts similar to the ones used in the lecture slides to illustrate this or a table). List the times for each context switch in your illustration. You can assume that the time slice is 15 milliseconds (if you were to need this).

[2 marks]

ii. Calculate the average response time for the priority queue algorithm.

[2 marks]

iii. Calculate the average turnaround time for the priority queue algorithm.

[2 marks]

**Question 2 (Concurrency and Memory Management)**

**[25 marks]**

a) List one advantage and one disadvantage for each of the following memory management schemes:

- Mono-programming
- Fixed partitions with equal size
- Dynamic partitions
- Paging
- Virtual memory with paging

[5 marks]

b) List and briefly explain the three conditions that a solution to the critical section problem must satisfy.

[4 marks]

c) Assume that you have a multi-core machine, on which two processes are running in parallel. Both processes access the same shared variable. The critical section that manipulates the shared variable is very short (e.g., only one instruction). Would you prefer a solution for the mutual exclusion problem based on busy waiting, or one based on semaphores? Briefly explain your answer.

[4 marks]

d) Assume you have two processes, P1 and P2. P1 has a high priority, P2 has a low priority. P1 and P2 have one shared semaphore (i.e., they both carry out waits and posts on the same semaphore). The processes can be interleaved in any arbitrary order (e.g. P2 could be started before P1).

i. Explain the problem with priority inversion

[1.5 marks]

Briefly explain whether the processes could deadlock when:

ii. both processes run on a Linux system as time sharing tasks

[1.5 marks]

iii. both processes run on a Windows 7 system as variable tasks

[1.5 marks]


continued on next page

iv. both processes run on a Windows 7 system as real-time tasks.

[1.5 marks]

e) For the code below, what will be the value for iCounter printed on the screen? Briefly explain your answer.

[3 marks]

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

#define NUMBER_OF_PROCESSES 2

int main() {
        pid_t pid = 0;
        pid = fork();
        int iCounter = 0;
        if(pid < 0) {
                printf("Could not create process\n");
                exit(1);
        } else if(pid == 0) {
                int i;
                for(i = 0; i < 10000; i++)
                        iCounter++;
                exit(0);
        }
        printf("The value of iCounter: %d\n", iCounter);
}
```

f) Assuming a 16 bit address space and the page table listed below, what are the physical addresses for the following logical addresses: 1234, 12288, 25810.

| Pages | | Frames | |
|---|---|---|---|
| 0 | 0000 | 0010 | 2 |
| 1 | 0001 | 0001 | 1 |
| 2 | 0010 | 0110 | 6 |
| 3 | 0011 | 0000 | 0 |
| 4 | 0100 | 0100 | 4 |
| 5 | 0101 | 0011 | 3 |
| 6 | 0110 | X | X |
| 7 | 0111 | X | X |
| 8 | 1000 | X | X |
| 9 | 1001 | 0101 | 5 |
| 10 | 1010 | X | X |
| 11 | 1011 | 0111 | 7 |
| 12 | 1100 | X | X |

[3 marks]

**Question 3 (Memory Management and Deadlocks)**

**[25 marks]**

a) Briefly explain the principle behind address relocation. Why it is a necessity on modern day interactive/multi-tasking machines. Explain how address relocation works in virtual memory with paging. Include an illustration.

[5 marks]

b) List and briefly explain the 4 conditions that must hold for deadlocks to occur (known as Coffman's conditions). Give two examples of how you could undermine these conditions to prevent deadlocks from happening.

[4 marks]

c) When creating a child process in Linux using `fork()`, would the child process have its own page table, or would its page table be shared with the parent process. Briefly explain your answer.

[3 marks]

d) Assume a machine that has a 42-bit virtual address space, and a 32-bit physical address space (i.e., the maximum amount of physical memory for a 32-bit address space is present).

i. How many entries would you expect a single level page table to have when the page size is 4 kilobytes? Briefly explain your answer.

[2 marks]

ii. How many entries would you expect to find in an inverted page table? Briefly explain your answer.

[2 marks]

iii. Assuming that the memory access time is 90ns and the time it takes to search the translation lookaside buffer is 15ns. What is the average memory access time for a page table with 3 levels and a 98% hit rate (write down in equations how you get the result)?

[2 marks]


continued on next page

e) Use and illustrate (by listing the steps in the execution of the algorithm) the banker's algorithm for multiple resources to determine the minimum value for **X** in the available resource vector below so that the state for the system given below is safe (explain your working).

The available resource vector is given by A = (1 0 **X** 1 1) (each column corresponds to a resource). The currently assigned resources are given by:

| | *R1* | *R2* | *R3* | *R4* | *R5* |
|---|---|---|---|---|---|
| *Process A* | 2 | 0 | 2 | 1 | 1 |
| *Process B* | 3 | 0 | 1 | 1 | 0 |
| *Process C* | 2 | 1 | 0 | 1 | 0 |
| *Process D* | 2 | 1 | 1 | 1 | 0 |

The maximum required resources are given by:

| | *R1* | *R2* | *R3* | *R4* | *R5* |
|---|---|---|---|---|---|
| *Process A* | 2 | 1 | 4 | 3 | 2 |
| *Process B* | 3 | 2 | 2 | 1 | 0 |
| *Process C* | 3 | 1 | 3 | 1 | 0 |
| *Process D* | 2 | 1 | 2 | 2 | 1 |

[4 marks]

f) Consider a swapping system for which the free list indicates the following blocks of memory (a) 10KB, (b) 15KB, (c) 12KB, (d) 21KB, (e) 30KB, (f) 9KB, (g) 16KB (in that order). Assume that requests for blocks of memory of 12KB, 9KB and 25KB come in (in that order). Which blocks would be allocated for the first fit, next fit, and best fit algorithm?

[3 marks]

**Question 4 (Memory Management, File Systems)**

**[25 marks]**

a) Briefly explain what thrashing is. What are the possible causes? How could thrashing be prevented?

[4 marks]

b) List one advantage and one disadvantage for each of the following file systems:
   i. Contiguous
   ii. Linked list
   iii. File allocation table
   iv. I-nodes

[4 marks]

c) Explain how the Linux file system (ext3 and ext4) improves performance and reliability over the standard Unix i-node file system.

[4 marks]

d) Assume that you have a file system that uses a rotational disk and on which the information will be written only once. The exact size of the files is known at the time of writing. What type of file system would be most appropriate in this case? Briefly explain your answer.

[3 marks]

e) Assume a page access time of 150ns, a disk rotating at 7200 RPM, an average seek time of 25ms, and an average page fault rate of 0.001. What would be the average memory access time and relative slowdown? Write down how you get to this result, i.e. the equations.

[3 marks]

f) Assume that you have a process with 8 logical pages and a system with 4 physical frames. Considering the order in which the pages are referenced given below, how many page faults would be generated by the least recently used page replacement algorithm? Please include the steps of your working.

```
Page references: 6 7 2 5 6 3 5 1 5 7 3 3
```

[4 marks]

g) Consider a disk with 300 tracks, and the following sequence of requested tracks 80, 150, 230, 50, 220, 50, 190, 260, 210, 270. Assume that the disk arm is located at position 60 and is moving in the up position (i.e. 1 to 300). Calculate the number of tracks crossed if using the `shortest seek time first` disk scheduling algorithm.

[3 marks]