University of Nottingham
UK | CHINA | MALAYSIA

Operating Systems and Concurrency

Lecture 24: Revision

University of Nottingham,
Ningbo China, 2024

COMP2046-E1

University of Nottingham
UK | CHINA | MALAYSIA

School of Computer Science

A Level 2 Module COMP2046 Autumn 2024/25

Operating System and Concurrency

Instructions:

Time allowed: TWO Hours

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

Candidates may complete the front cover of their answer book and sign their desk card. **DO NOT turn the examination paper over until instructed to do so**.

Permitted resources:

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.

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

Prohibited resources:

All other dictionaries, including subject specific translation dictionaries and electronic dictionaries. Electronic devices capable of storing and retrieving text.

Appended material: NONE

Additional material: NONE

Information for Invigilators: NONE

Exam Papers must not be removed from the examination venue.

OSC Exam questions

- This is a two-hour exam, four main questions.
  - Process, Threads and Scheduling [25 marks]
  - Concurrency and Deadlock [25 marks]
  - Memory Management [25 marks]
  - File systems [25 marks]
- Each of which comes, with 4-5 sub-questions
- All the questions have been derived form the **lectures and labs**
- Past exam papers are available on the Moodle.

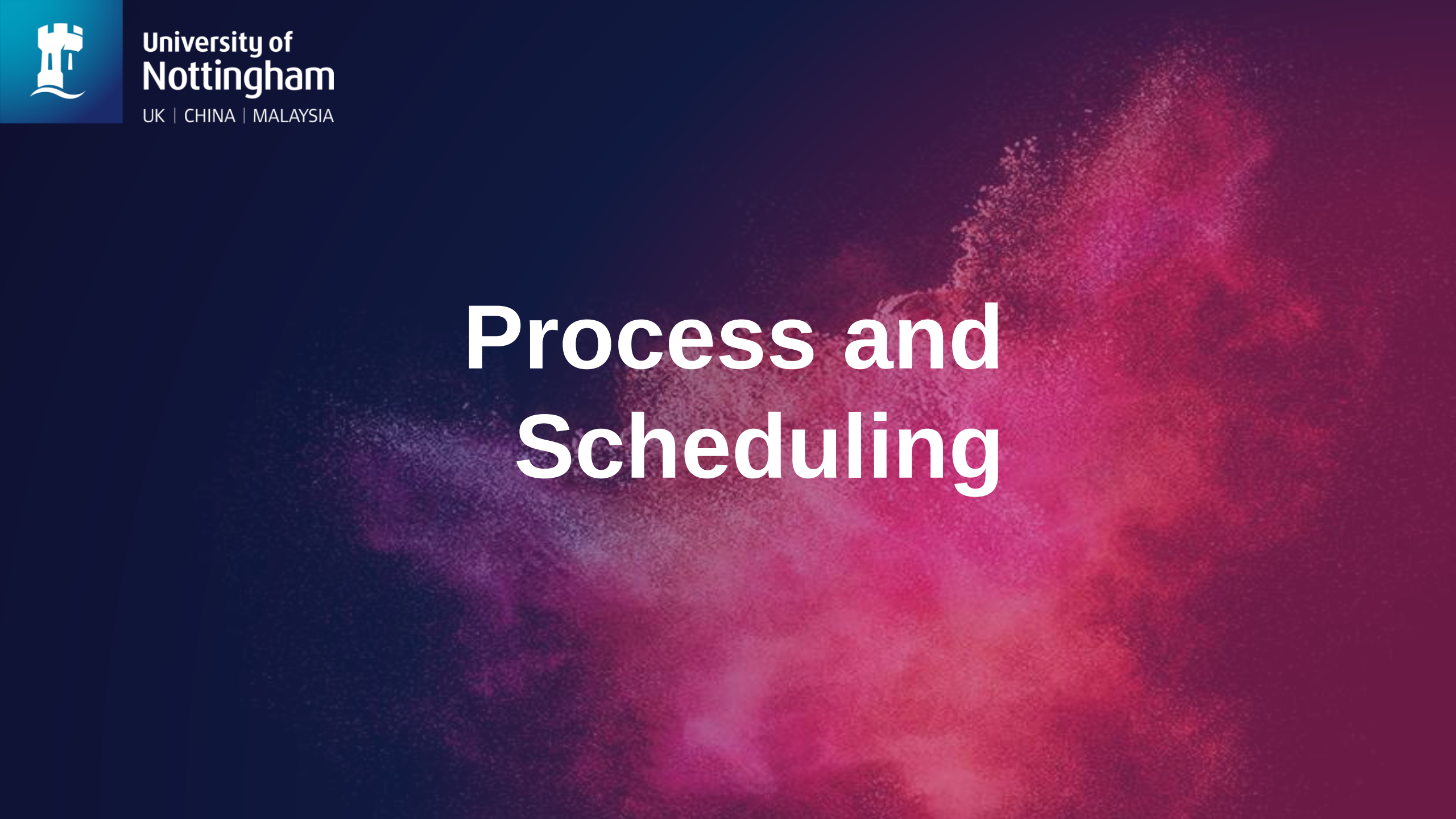
OSC Exam questions

- The exam focusses on:
  - **Knowledge:** ability to memorize and to recall terms, facts and details.
  - **Comprehension:** ability to summarize and describe in your own words.
  - **Application:** to apply or transfer learning to their own life or to a context different than one in which it was learned.

- The exam consist of different type of questions:
  - Short answer and Essays
  - Computational
    - Computational questions require that students perform calculations in order to solve for an answer.
  - Write program and providing output

OSC Exam questions

- Make sure you know **how to apply every algorithm** that was covered in the lectures (e.g. process scheduling, deadlock avoidance, page replacement, disk scheduling, etc.)
- **Focus on the question** in your answers:
  - i.e. avoid adding info that is not asked for – and subsequently run out of time
- Write down equations (i.e., how do you get the results) and bring a calculator
- Pay attention to your **handwriting** :-)


University of
Nottingham
UK | CHINA | MALAYSIA
Process and
Scheduling

Process and Scheduling

- Covers the following, but not limited to:
  - OS definitions, responsibilities and structures/architecture
  - Process, process states and transitions
  - Process scheduling
    - First Come First Served (FCFS)(Batch System)
    - Shortest job first (Batch System)
    - Round Robin (Interactive System)
    - Priority queues (Interactive System)
  - Thread definition and implementations (One to one, Many to one, Many to Many)
  - System calls and POSIX thread libraries we that we've discussed in the lab

Process and Scheduling – Application

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]

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

[2 marks]

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

[2 marks]

Tips [application] (lecture 4)

- To answer this question:
  - 1- use Gantt charts similar to the lectures, how the processes start and when they finish according to the scheduling algorithm (see lecture 4 examples)
  - 2- Calculate the average RT and/TT according to the algorithm (see lecture 4)

Process and Scheduling – Comprehension (lecture 5 and 6)

- *"In which situations would you **favour user level threads**? In which situations would you **favour kernel level threads**. Explain your answer." [2 marks]*

  - User threads run entirely in **user space** and rely on user libraries. These are useful when the **underlying operating system does not provide thread support**, or is **a single CPU machine**
  - User level threads do not support **true parallelism in multicore/multiprocessor** architectures. Kernel level threads do enable to use true parallelism

Process and Scheduling – Application (labs)

Provide the correct output(s) for the code below. [4 marks]

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

int value = 10;
int main()
{
pid_t pid = fork();
int status;
if (pid == 0) {
 sleep(10);
 value += 20;
 execlp("/bin/echo","echo","Hello from the child process",NULL);
 printf("Child: Valie = %d\n", value);
 return 0;
}
printf("Hello from the parent process\n");
waitpid(pid, &status, 0);
printf("Parent: value = %d\n", value);
return 0;
}
```

```
Hello from the parent process
Hello from the child process
Parent: value = 10
```

Concurrency and deadlocks

Concurrency and deadlock

- Covers the following, but not limited to:
  - Concurrency definitions and problems
  - Race condition
  - Critical section and critical section problems(Mutual exclusion, Progress & Bounded waiting)
  - CS problem solutions
    - Software (Peterson solution), hardware (Disable interrupt, Test and set lock, Compare and Swap, and operating system based (semaphore, mutex, monitor)
  - Deadlock, definition/condition(mutual exclusion, no preemption, hold & wait and circular /detection/recovery
  - Deadlock avoidance, banker algorithms

Concurrency and Deadlock- Knowledge/comprehension

Explain how Binary Semaphores can fulfill the three critical section problem requirements. Provide pseudo-code examples to illustrate how this mechanism could potentially lead to a deadlock.

[6 marks]

Semaphore which works with two atomic functions: wait and signal. The initial value is 1.

.....

Wait(S)

CS

Signal(S)

Concurrency and Deadlock: Application

- Tips [lab 7]

A coffee shop has several self-service kiosks where customers can place their coffee orders. Each kiosk can handle one customer at a time. The total number of kiosks is K. If all kiosks are in use, arriving customers must wait for a kiosk to become free. When a customer finishes placing their order, the next customer in line is allowed to use the kiosk. Using semaphores, write **pseudocode** for both the order process/thread and the completion process/thread, ensuring that the kiosks are efficiently managed. [5 marks]

Concurrency and Deadlock: Application [Tips lab 7]

Kiosk_sem = k

```
// Function to signal that the kiosk is now free
function free_kiosk(void *arg) {
    int customer_id = *(int *)arg;
    printf("Customer %d has finished placing the order and leaves the kiosk.\n",
customer_id);
    // Signal that the kiosk is now free
    sem_post(&kiosk_semaphore);
}
```

```
// Function for a customer placing an order
function place_order(void *arg) {
    int customer_id = *(int *)arg;
    printf("Customer %d arrives and tries to use a kiosk.\n", customer_id);
    // Wait for a kiosk to be available
    sem_wait(&kiosk_semaphore);
    Self_service() // Simulate the place an order
    printf("Customer %d is using a kiosk to place an order.\n", customer_id);
    free_kiosk()
}
```

Concurrency and Deadlock- application

- Tips [This example taken from Lecture 14]

Consider a system with five processes P0 through P4 and three resource types A, B, and C.
Resource type **A has ten** instances, resource **type B has five** instances, and resource type **C has seven** instances.
Suppose that, at time T0, the following snapshot of the system has been taken.

- What will be the content of the Need matrix? [2 marks]
- Is this state safe? [2 marks]
- What will happen if process P1 requests **one additional** instance of resource **type A** and **two instances** of resource **type C**? [2 marks]

	Allocation			Max			Need			Available		
										3	3	2
P0	0	1	0	7	5	3						
P1	2	0	0	3	2	2						
P2	3	0	2	9	0	2						
P3	2	1	1	2	2	2						
P4	0	0	2	4	3	3						


University of
Nottingham
UK | CHINA | MALAYSIA
Memory
Managment

Memory Management

Covers the following, but not limited to:

- Memory management and mono-multi programming
- Code relocation and protection
- Paging and address translation
- Page replacement algorithms
- Large page table management
- Thrashing

Memory Management – knowledge

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

Memory Management – application

- 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

Memory Management – application

- 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 partitions would be allocated for the first fit, next fit, and best fit algorithm?


University of
Nottingham
UK | CHINA | MALAYSIA
File systems

File Systems

Covers the following, but not limited to:

- Disk construction, access and scheduling
- File systems' user and implementation view
- File system structures and directories
- Free-space managements
- Implementations of File systems
- File system paradigms

File systems- Knowledge and Comprehension

Define symbolic and hard links, compare them and outline their advantage and disadvantage for file sharing. Explain which one offer better benefits to distributed systems.

[3-4 marks]

Tips [application] (lecture 17)

You need to provide discussions and details to explain the topics.

- Define clearly hard/soft links.
- Outline the dis/advantages for both
- Compare dis/advantage
- Explain which one offers better benefits to distributed systems

File systems- comprehension

Explain how i-node addresses the issues of the FAT file system.

[3 marks]

Tips

You need to provide discussions and details to explain the topics.

- Define both FAT and I-node
- Outline the dis/advantages for both
- Compare dis/advantage and explain how i-node address the issue of FAT file system

File systems- Application

For a 10,000 RPM disk, with 300-sector tracks and track change (seek) time 800 µsec, assuming no cylinder skew? Calculate access time for a file (contiguous or random) with 3000 sectors.

[4-5 marks]

Tips

To calculate access time:
Depending to the format of the file access (contiguous or random) the access time is measured.

- For contiguous access
  - The first track takes: seek + rotational delay + transfer time
  - Other track take: rotational delay + transfer time
- For random
  - All sectors requires: seek + rotational delay + transfer time

University of Nottingham
UK | CHINA | MALAYSIA

Good Luck with your Exam!