

DUBLIN INSTITUTE OF TECHNOLOGY  
KEVIN STREET DUBLIN 8

---

## BSc. (Honours) Degree in Computer Science

Year 2

---

### Semester 1 Examinations 2013/2014

---

#### Operating Systems 2

Mr. Brian Gillespie  
Dr. D. Lillis

Monday 13<sup>th</sup> January

1.00 p.m. – 3.00 p.m.

Question 1 is **compulsory**.  
Answer question 1 **and** two of the other three questions.

1. (a) In a multi-threaded environment, a thread transitions between various states during its lifetime. Describe three such thread states and explain what it means for a thread to be in each one. A thread may also live on one or more system queues during its lifetime. Describe two kinds of thread queues and what each is used for. (10 marks)
- (b) In a POSIX operating system, processes are created by the `fork()` system call. Illustrate how this mechanism works using Python or C code. A call to `fork()` is said to return in two places. Explain what this means. A call to `fork()` may also fail. Give one reason why this may occur. (10 marks)
- (c) Describe each of the following POSIX inter-process communication mechanisms:
- i. Pipes
  - ii. Shared Memory
- For each one, give an appropriate example of where it can be used. (10 marks)
- (d) What is a *critical region* in the context of an asynchronous operating system? Describe two synchronisation mechanisms you have studied and show how you would use each in C or Python (10 marks)
- (e) The POSIX command line interface (shell) is said to be highly composable. Explain what this means. Describe, with examples, how processing and filtering operations are carried out on line-oriented text data in a POSIX shell (10 marks)
2. (a) What is a REPL in the context of operating system user interfaces? (5 marks)
- (b) Describe how a UNIX shell executes external programs in a POSIX environment. What is the function of the `PATH` environment variable in this mechanism? (10 marks)

- (c) Explain how a shell can implement standard input and standard output redirection when executing commands, illustrating your answer with relevant C or Python code.

(10 marks)

3. (a) In the Pintos operating system, there are two data structures stored in the first page of thread memory. What are the functions of each?

(5 marks)

- (b) Describe, using the Pintos operating system, how the next ready-to-run thread is scheduled onto the CPU when the currently running thread yields or becomes blocked. Why does part of this mechanism need to be implemented in Intel 80x86 assembly language?

(10 marks)

- (c) Outline one possible implementation of a fair, priority-based thread scheduler, using a multi-level feedback queue. How could the implementation guard against lower priority threads being permanently starved of CPU time by higher priority threads?

(10 marks)

4. (a) Explain what is meant by a race condition in asynchronous systems?

(5 marks)

- (b) Describe, in detail, a scheme for transmitting messages between two or more threads using a fixed-sized ring buffer. Illustrating your answer with Python or pseudo-code, show the principle elements of your answer, including the necessary synchronisation primitives for a correct implementation.

(10 marks)

- (c) Describe, how the standard user authentication mechanism is implemented in UNIX, including the mechanism by which a user password is verified.

(10 marks)