UNIVERSITY COMPUTER SCIENCE • EXERCISE SHEET

7.1 Processes and threads

Name: _____ Class: _____ Date: _____ **Total: 24 marks**

KEY WORDS process • context switch

Objective

Trace process states and price the two kinds of switch.

1 Key knowledge and worked examples

■ The states

New, ready (waits only for the CPU), running, blocked (waits on a device), terminated ✓. Dispatch and preempt move ready ↔ running; block and wake connect running and blocked ✓.

■ Worked example: the switch

The PCB holds registers, PC, page-table pointer, open files – 2000 bytes at 2 GB/s costs 1 µs to save and restore, plus cache pollution ✓.

2 Practice

2.1 A process waits for a disk read. Name its state, and the state it enters when the read completes. [3]

2.2 Name the three things a thread owns that its process does not re-own per thread. [3]

2.3 Compute the PCB switch time for 5000 bytes at 2.5 GB/s. [3]

3 Exam-style questions

3.1 A process runs on a machine where the timer interrupts every 10 ms; a disk read it issues takes 4 ms, and the switch costs 1 µs.

Trace the state changes from the moment the read is issued until the process runs again, naming each transition and its cause. Compute the CPU time lost to the switch as a percentage of the 4 ms wait. State why the faulting instruction can simply restart, and what would be lost if it could not. [8]

3.2 A server handles each connection with a thread (create plus switch 50 µs) or a process (500 µs).

Compute the setup cost for 20 connections under each design. Give one shared resource that makes threads cheaper to run, and the single failure that makes them riskier. State which design isolates a crash and why. [7]

Solutions

2.1

- Blocked; it enters ready when the read completes ✓.

2.2

- Its own stack, its own registers, its own program counter ✓.

2.3

- $5000/2.5 \times 10^9 = 2$ µs ✓.

3.1

- Running → blocked (read issued) ✓.
- Scheduler dispatches another process (ready → running elsewhere) ✓.
- Read completes: blocked → ready (wake) ✓.
- Dispatcher picks it: ready → running; the instruction restarts ✓.
- Switch cost 1 µs of a 4 ms wait = 0.025% ✓.
- The instruction is restartable because the fault happens before any visible side effect; otherwise preemption would lose or duplicate work ✓.

3.2

- Threads: 20×50 µs = 1 ms ✓.
- Processes: 20×500 µs = 10 ms ✓.
- Shared: the address space (code and cache) – no per-connection copy ✓.
- Riskier: one thread's crash takes the whole process – one address space ✓.
- Processes isolate: separate spaces mean a crash cannot corrupt another connection's state ✓.