Solutions

Each answer shows the steps, then the **result**.

2.1

- any three, e.g. process, memory, file or device management, security, user interface

2.2

- (i) A at **3**, B at **8**, C at **10**
- (ii) A 0–2, B 2–4, C 4–6 (ends), A 6–7 (ends), B 7–10 (ends): C **6**, A **7**, B **10**

2.3

- the handler uses the registers too, so their values would be lost
- saving them lets the interrupted program carry on exactly where it stopped

3.1 B

- each program has its own protected area

3.2 B

- starvation

3.3 B

- rare and unpredictable

3.4

(a)

- the scheduler gives each app a short time slice of the CPU in turn
- it switches so fast that all three seem to run together

(b)

- game 0–2, music 2–4 (ends), messaging 4–6, game 6–8, messaging 8–9 (ends), game 9–11 (ends)
- music **4 ms**, messaging **9 ms**
- game **11 ms**

(c)

- the CPU finishes its current instruction and saves its registers
- it runs the interrupt handler for the message
- it restores the registers and the game carries on

(d)

- music must play without gaps: a late slice is heard as a glitch
- a slightly slower game frame is less noticeable