2005 Operating Systems Foundations (solution) draws on parts 1,2,3 of 10 & 6 JMB

(a) (i) Devices (and humans) operate far more slowly than processors.
1 eg. Many millions of instructions can be executed while a character is being typed.

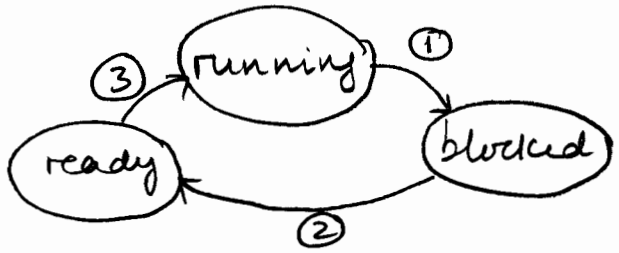
(ii) UART typical registers: data-in | status (is data-out free for next char.? does data-in contain a char.?) | data-out | control ... includes interrupt enable.

(iii) The software checks the status bits in the UART periodically
- is there a character in data-in?
- is data-out free for the next output character?

2 checking period – likely to cycle around all polled devices and be time-driven.

(iii) The UART sends an interrupt signal to the processor – recorded in a CPU register. When this becomes of highest priority (c.f. interrupt level at which CPU is executing) there is an
7 automatic transfer of control to the ISR. PC & CPU status are saved (& restored).

(iv)



① the process must wait until the next char is ready on input or the device is ready on output. Process marked as blocked in PSD and event awaited is recorded. Process removed from ready queue & recorded as awaiting event in event structure (system-dep.)

② character (or device) ready & UART signals interrupt. ISR records process status as ready in PSD and adds process to ready queue.

③ OS scheduler selects process to run.

(b) (i) Independently scheduled processes require access to shared data
2 The data structures would be corrupted by uncontrolled access.

(ii) A process acquires access to the buffer then finds it cannot proceed eg. buffer full on input – driver waits for app. to remove
eg. " " " output – app. ---- driver " "
eg. buffer empty on output – driver waits for app to insert
2 " " " " – app ---- driver " "

(iii) The driver first acquires buffer-lock then tests the state of the buffer. If it finds it cannot proceed it WAITS for the required condition but does not first release buffer-lock.
4 DEADLOCK – both processes are blocked indefinitely.
The semaphores must also be signalled.
The buffer lock/unlock should be around the read & write only.