

Esercizi su Gestione Input/Output, in "busy wait" e con interrupt

Esercizio su "Busy Wait":

Integrare il codice del file "esGetCh.s", che contiene:

- due procedure GetCh e PutCh che interagiscono con i dispositivi tastiera e video:
 - GetCh riceve un carattere dal dispositivo Receiver (tastiera), restituendolo in \$v0
 - PutCh invia un carattere al dispositivo Transmitter (video), ricevuto in \$a0
- due procedure ReadStre e WriteStr che invocano GetCh e PutCh:
 - ReadStre usa GetCh e memorizza una stringa di caratteri, che termina con l'invio (non memorizzato!), a partire dall'indirizzo di memoria ricevuto in \$a0
 - WriteStr usa PutCh per inviare a video una stringa, che preleva a partire dall'indirizzo di memoria ricevuto in \$a0; la fine stringa in memoria coincide con un byte contenente 0
- un main che, usando le procedure sopra descritte::
 - legge una stringa
 - la invia a video
 - invia un a capo
 - esce quando riceve la stringa vuota.

Integrazioni richieste:

- implementare GetCh, mutuandolo eventualmente da PutCh, facendo attenzione alla scelta dei registri da usare ed eventualmente salvare
- inserire dove opportuno delle chiamate a PutCh e/o WriteStr in modo da avere:
 - un echo carattere per carattere di quanto si digita
 - un echo dell'intera stringa gia' all'interno di ReadStre.

Esercizio per I/O con gli interrupt:

N.B. In Spim disabilitare il caricamento di qualunque gestore eccezioni.

Integrare il codice "interruptGetCh.s", che contiene:

- il codice del precedente esercizio, modificato per operare con gli interrupt
- il gestore di eccezioni: si occupa solo degli interrupt dei due dispositivi SPIM video e tastiera; ad ogni interrupt di tastiera visualizza su console un punto, e ad ogni interrupt video visualizza una virgola
- un piccolo blocco di istruzioni per inizializzare opportunamente gli interrupt in CPU e dispositivi; queste istruzioni iniziano in corrispondenza dell'etichetta __start
- un blocco PutCh che, operando con video+interrupt, invia direttamente a video se il dispositivo e' ready, altrimenti memorizza il carattere nel buffer associato all'etichetta aVideo.

Integrazioni richieste:

- implementare GetCh, che sara' pero' molto diverso dall'attuale PutCh: anche in questo caso vi sara' in pratica un ciclo che continuera' ad esaminare il buffer tastiera associato all'etichetta daTast fino a che apparira' disponibile un carattere...
- inserire dove opportuno un ciclo di pausa in modo che tutti i caratteri digitati escano a video, e non solo alcuni (cercando di capire perche' alcuni altri non escono nel codice senza pause)
- inserire dove ritenuto utile istruzioni che producano echo di cio' che si digita, e vedere l'effetto dell'interazione tra interrupt della tastiera e del video.

VARIANTE a questo esercizio:

Integrare il codice "interruptGetCh2.s", che contiene:

- tutto "interruptGetCh.s", ma con il gestore eccezioni modificato per saltare DENTRO GetCh, all'etichetta ChDisponibile, quando riceve un carattere (NB: per semplificare l'implementazione questo accadrà qualunque cosa fosse in esecuzione prima dell'interrupt).

Integrazione richiesta:

- ora GetCh, quando non è disponibile un carattere, invece di attendere in modo busy wait deve eseguire del codice utile, già presente nel codice, quindi si deve:
- completare GetCh in modo che se c'è un carattere lo restituisca subito al chiamante, e altrimenti passi ad eseguire il "codice utile" (già presente).