

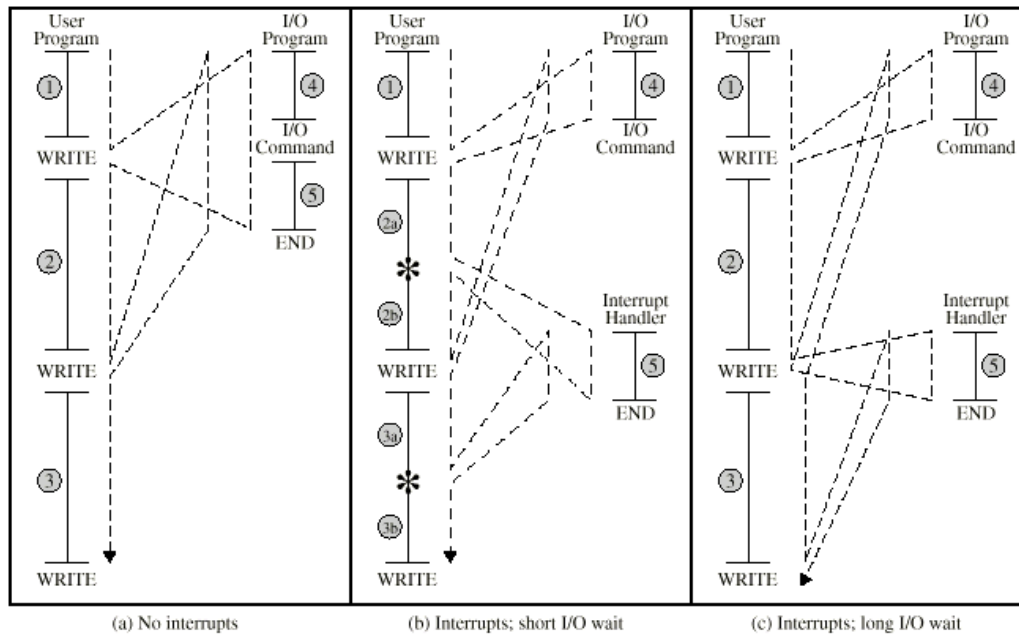
Interruzioni

- Meccanismo tramite il quale altri moduli (esempio I/O) possono interrompere la normale sequenza di esecuzione
- Tipiche interruzioni
 - Program
 - Esempio: overflow, division by zero
 - Timer
 - Generata da un timer interno alla CPU
 - I/O
 - Per segnalare la fine di un'operazione di I/O
 - Guasto hardware
 - Esempio: mancanza di alimentazione

Perché interrompere?

- Per migliorare l'efficienza della elaborazione
- Esempio:
 - Molti dispositivi esterni sono più lenti del processore
 - Per evitare che la CPU attenda la fine di un'operazione di I/O

Esempio



Trasferimento del controllo per una interruzione

USER PROGRAM

```

<statement>
<statement>
⋮
<statement>
    } Code segment 1

```

WRITE

```

<statement>
<statement>
⋮
<statement>
    } Code segment 2

```

WRITE

```

<statement>
<statement>
⋮
<statement>
    } Code segment 3

```

I/O PROGRAM

```

<statement>
<statement>
⋮
<statement>
    } Code segment 4

```

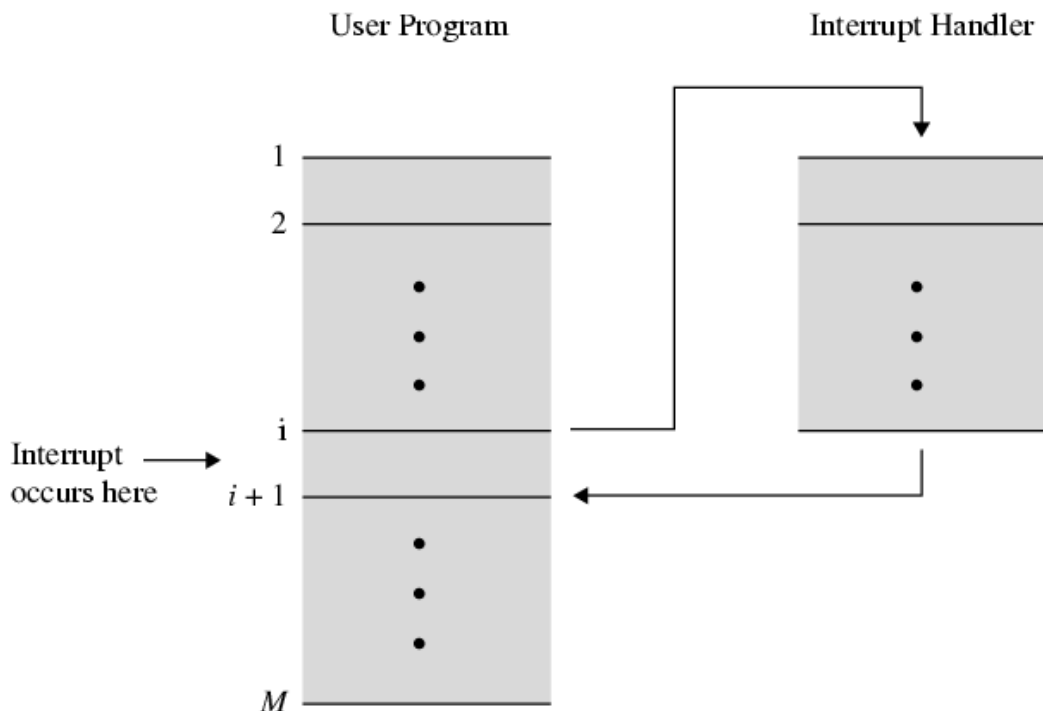
I/O command

```

<statement>
<statement>
⋮
<statement>
    } Code segment 5

```

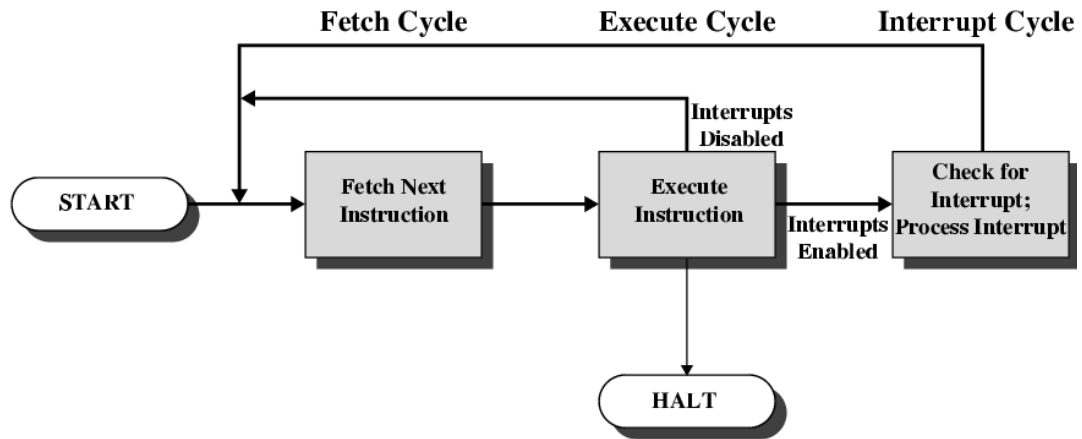
Trasferimento del controllo per una interruzione



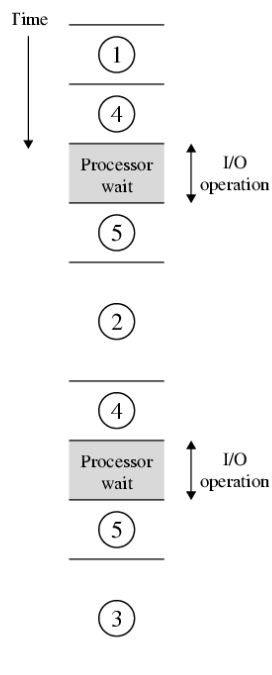
Ciclo di interruzione

- Aggiunto al ciclo di esecuzione
- La CPU controlla se ci sono interruzioni pendenti
- Se no, prende la prossima istruzione
- Se si:
 - Sospende l'esecuzione del programma corrente
 - Salva il contesto (es.: indirizzo prossima istruzione)
 - Imposta il PC all'indirizzo di inizio del programma di gestione dell'interruzione
 - Esegue il programma di gestione dell'interruzione
 - Rimette il contesto al suo posto e continua il programma interrotto

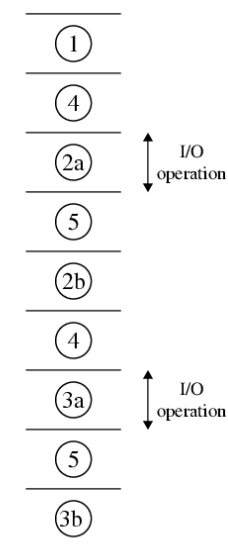
Ciclo di esecuzione con interruzioni



Breve attesa di I/O



(a) Without interrupts

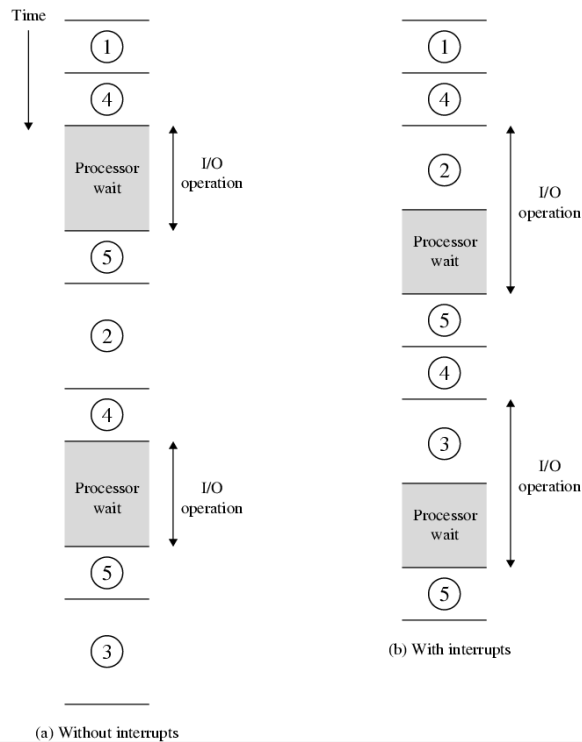


(b) With interrupts

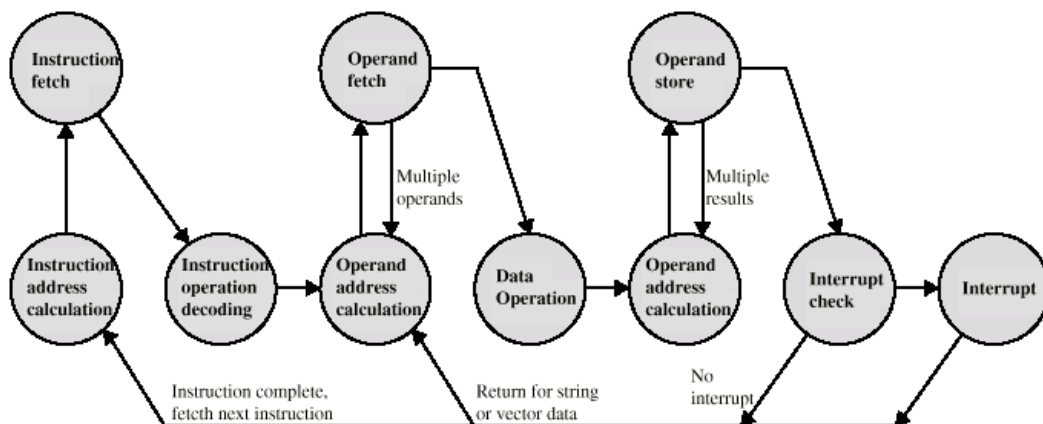
Tempo per operazione di I/O minore del tempo tra due istruzioni WRITE

Le interruzioni eliminano le attese

Lunga attesa di I/O



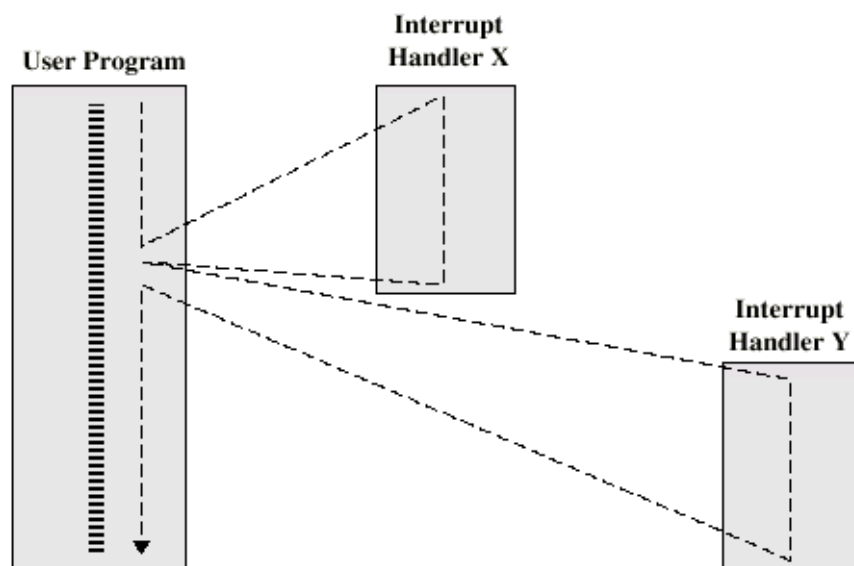
Ciclo con interruzioni



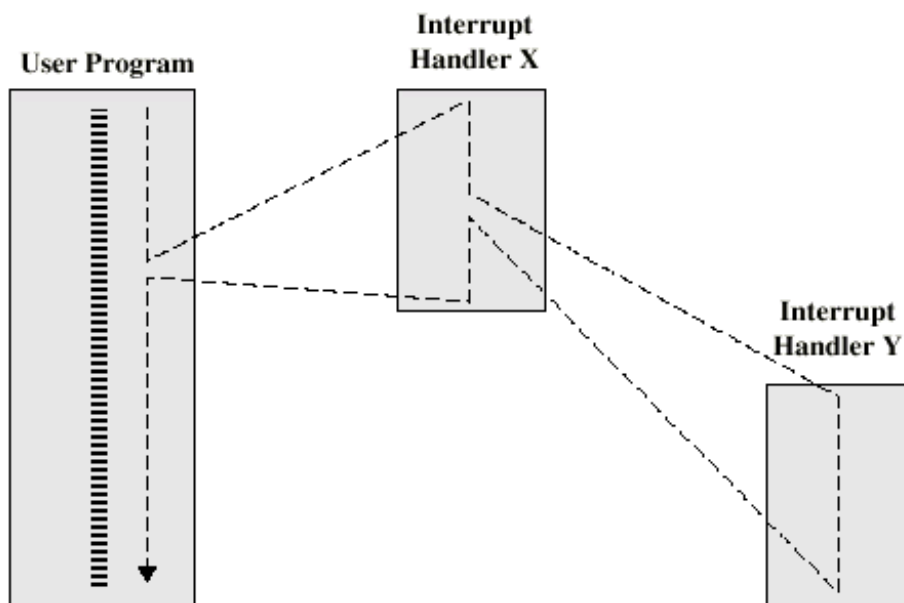
Interruzioni multiple

- Disabilitare le interruzioni
 - La CPU ignorerà altre interruzioni mentre gestisce la prima
 - Le interruzioni rimangono pendenti e sono controllate solo dopo che la prima è stata gestita completamente
 - Interruzioni gestite nella sequenza in cui sono richieste
- Definire delle priorità
 - Interruzioni con bassa priorità possono essere interrotte da interruzioni con priorità più alta
 - Quando l'interruzione con priorità più alta è stata gestita, la CPU ritorna all'interruzione precedente

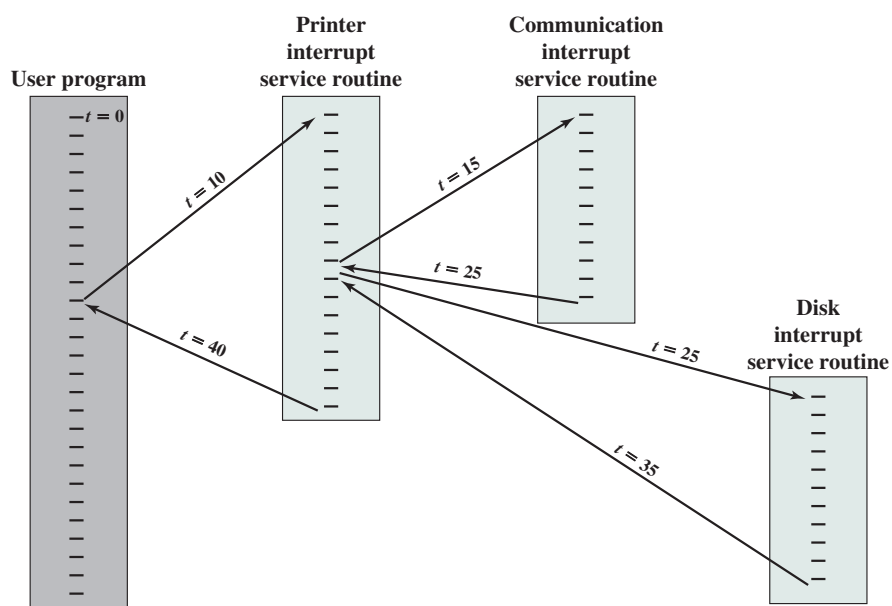
Interruzioni multiple – sequenziali



Interruzioni multiple – annidate

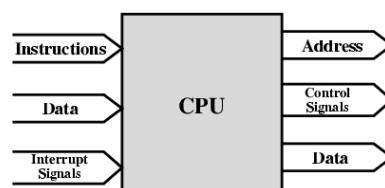
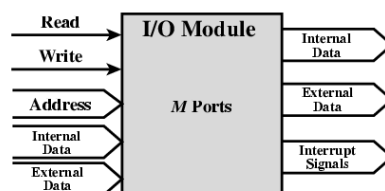
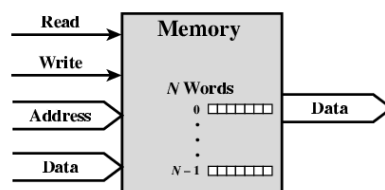


Esempio di Interruzioni



Connessioni

- Tutte le componenti di un calcolatore devono essere connesse
- Tipi diversi di connessione per diversi tipi di componente
 - Memoria
 - Input/Output
 - CPU





Connessioni per la memoria

- Riceve e spedisce dati (scrittura e lettura)
- Riceve indirizzi (di locazioni di M)
- Riceve segnali di controllo
 - ☐ Lettura
 - ☐ Scrittura



Connessioni dell' Input/Output (1)

- Modulo di I/O: simile ad una memoria dal punto di vista della CPU
- Operazioni di Output
 - ☐ Riceve dati dalla CPU
 - ☐ Manda dati alle periferiche
- Operazioni di Input
 - ☐ Riceve dati dalle periferiche
 - ☐ Manda dati alla CPU



Connessioni dell'Input/Output (2)

- Riceve segnali di controllo dalla CPU
- Manda segnali di controllo alle periferiche
- Riceve indirizzi dalla CPU (n.ro di porta per identificare una periferica)
- Manda segnali di interruzione



Connessioni per la CPU

- Legge istruzioni e dati
- Scrive dati (dopo l'elaborazione)
- Manda segnali di controllo alle altre unità
- Riceve segnali di interruzione



Connessioni

- Da M a CPU: la CPU legge un'istruzione o un dato dalla M
- Da CPU a M: la CPU scrive un dato in M
- Dall'I/O alla CPU: la CPU legge i dati di una periferica
- Dalla CPU all'I/O: la CPU invia dati ad una periferica
- Dall'I/O alla M o viceversa: accesso diretto alla M da parte di un dispositivo di I/O