Politecnico di Torino

Programmazione di Sistema

Anno accademico 2022/2023

Marco Lampis

28 febbraio 2023

Indice

0 Informazioni

La seguente dispensa è stata realizzata nell'anno accademico 2022-2023 durante il corso di *Tecnologie e Servizi di Rete*. Il materiale **non** è ufficiale e non è revisionato da alcun docente, motivo per cui non mi assumo responsabilità per eventuali errori o imprecisioni.

Per qualsiasi suggerimento o correzione non esitate a contattarmi o a eseguire una pull request su GitHub.

E' possibile riutilizzare il materiale con le seguenti limitazioni:

- Utilizzo non commerciale
- Citazione dell'autore
- Riferimento all'opera originale

E' per tanto possibile:

- Modificare parzialmente o interamente il contenuto

Questi appunti sono disponibili su GitHub al seguente link:

```
https://github.com/Guray00/polito_lectures
```

Figura 1: Repository GitHub

0.1 Contributi

La condivisione è alla base del successo di qualsiasi progetto, citando:

> *“Open source is about collaborating; not competing. ~ Kelsey Hightower”*

La seguente dispensa ha avuto il prezioso contributo di:

- Marco Lampis

1 Memory Management

Lo studio della gestione della memoria permette di capire come questa sia organizzata dal punto di vista hardware mediante varie tecniche. Si prenderà ad esempio un Intel Pentium in quanto supporta sia la segmentazione che la paginazione segmentata.

1.1 Introduzione

Ogni programma prima di essere eseguito deve essere caricato dalla memoria di massa (hard disk) in memoria principale (RAM) in modo da riservarne memoria per il processo. Solo dopo ciò il programma può essere eseguito.

Per far si che il programma funzioni, ogni dato utilizzato deve essere direttamente accessibile dalla memoria, che può richiedere tempistiche differenti a seconda della posizione:

- in un registro il tempo di accesso è di circa un clock (o meno).
- nella memoria principale possono essere richiesti più cicli di clock in quanto il dato potrebbe non essere disponibile e causare uno stallo.
- in cache, situata tra la memoria principale e i registri

1.2 Protection

La **protection** consente di limitare lo spazio di indirizzamento disponibile per un dato processo, in modo che questi non possano accedere reciprocamente alla memoria che gli viene riservata.

Un primo modo per implementare tale limitazione è mediante l'utilizzo di due registri denominati **base register** (indirizzo fisico minore) e **limit register** (dimensione del range), i quali stabiliscono i limiti in termini di indirizzi logici indirizzabili dal processo.

Con address binding si fa riferimento all'operazione di sistema che assegna a un indirizzo di memoria un programma o un processo a tempo di esecuzione.

Gli stage più importanti del ciclo di vita di un programma sono:

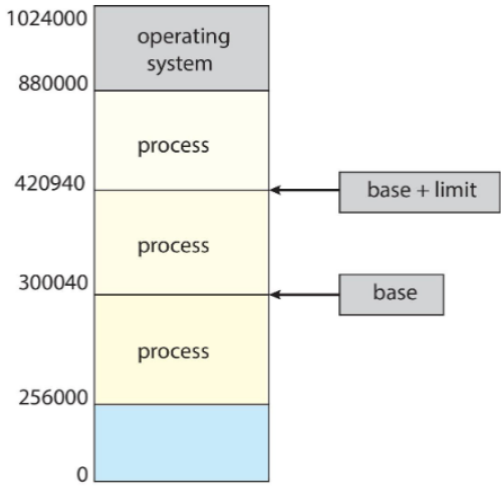


Figura 1.1: Limitazione dello spazio di indirizzamento

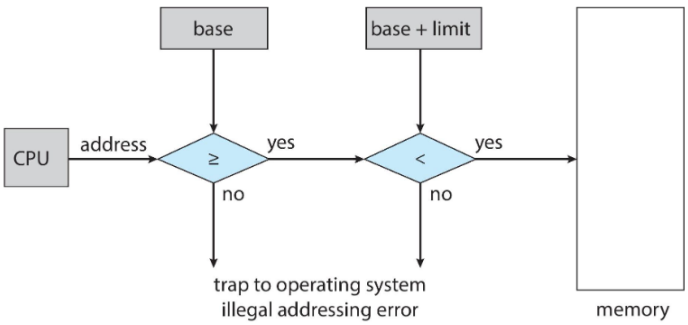


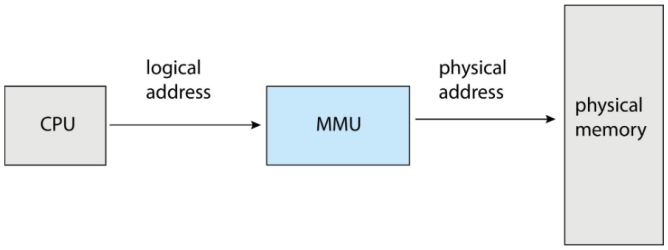
Figura 1.2: Algoritmo di verifica

- **Compile time**: la fase in cui il codice sorgente di un programma viene converito in codice macchina, ovvero un file eseguibile che può essere eseguito sul sistema.
- **Load time**: dopo la creazione dell'eseguibile, questo viene caricato in memoria ed eseguito.
- **Run time**: durante l'esecuzione il processo viene mosso da una segmento di memoria verso un altro segmento di memoria.

L'operazione di binding consiste nell'associare un indirizzo logico in un indirizzo fisico.

1.3 Memory Management Unit

Lo scopo della MMU è quello di porsi come intermediaro tra la CPU e la memoria fisica.



1.4 Allocazione contigua

…

1.5 Variable partition

…