

Layout (*Disposizione*) della Memoria Centrale (RAM) per un Processo (*Task*) → un programma in esecuzione

RAM



2. SEGMENTO DEL CODICE

- contiene le istruzioni del processo cioè del programma in esecuzione
- i registri **CS (Code Segment)** ed **IP (Instruction Pointer)** contengono inizialmente
- l'indirizzo di partenza del programma e durante l'esecuzione l'istruzione corrente
- il segmento ha una dimensione variabile (dipende da quanto è grande il programma)
- quando abbiamo un programma che occupa più spazio di quello disponibile si ricorre a tecniche di VIRTUALIZZAZIONE della MEMORIA

3. SEGMENTO DEI DATI

- contiene i dati globali (a cui posso avere accesso da qualsiasi punto del programma in esecuzione cioè ad esempio, relativamente ai programmi Ce C++, i dati dichiarati nel main o dopo gli include)
- ha una dimensione non maggiore di 64k

I registri : 1) DS =Data Segment } sono registri per indirizzare DATI
2) ES =Extra Segment }

4. STACK

- è una zona di dati
- usa i registri : 1) SS =Stack Segment
2) SP =Stack Pointer
- ha una dimensione fissa non maggiore di 64k
- usa il funzionamento di tipo LIFO (Last In First Out);
- questa zona dati mediante funzionamento FIFO è un segmento di appoggio per il passaggio di variabili ad un sottoprogramma e la gestione delle sue variabili locali
- l'immissione in STACK viene chiamata PUSH e la estrazione POP
- memorizza le variabili locali (cioè i dati contenuti da un sottoprogramma)

5. HEAP

- è la parte della RAM che rimane libera dopo aver disposto tutti i precedenti segmenti
- viene usata solo per le variabile dinamiche (fondamentali per il linguaggio Java)