

4 Gestione della memoria

4.1 Record di attivazione

Definizione 4.1 (Supporto a tempo di esecuzione). È l'insieme di strutture dati e funzioni necessarie all'esecuzione dei programmi e viene aggiunto al codice eseguibile dal compilatore.

Definizione 4.2 (Dynamic chain o call chain). Rappresenta la sequenza di chiamate e serve a garantire il corretto ordine di esecuzione. Implementa il passaggio del controllo in caso di chiamate annidate e tiene traccia dell'ordine.

Definizione 4.3 (Static chain). Implementa lo scoping statico e garantisce che i nomi siano referenziati rispettando la visibilità di variabili e funzioni.

Definizione 4.4 (Activation record o stack frame). Contiene tutte le informazioni necessarie all'esecuzione del blocco o della funzione.

Puntatore catena dinamica	Indirizzo del record di attivazione della funzione chiamante
Puntatore catena statica	Indirizzo del prossimo record di attivazione dove risolvere i nomi non presenti nel blocco corrente (implementazione dello scoping statico)
Indirizzo di ritorno	Indirizzo dell'istruzione da eseguire al termine della funzione/blocco corrente
Indirizzo risultato	Indirizzo nel record di attivazione del chiamante per memorizzare il risultato
Parametri	Spazio riservato alla associazione parametri formali - parametri attuali
Variabili locali	Spazio riservato alla allocazione delle variabili locali al blocco
Risultati temporanei	Spazio riservato alla allocazione delle variabili temporanee generate dal compilatore

4.2 Divisione della memoria

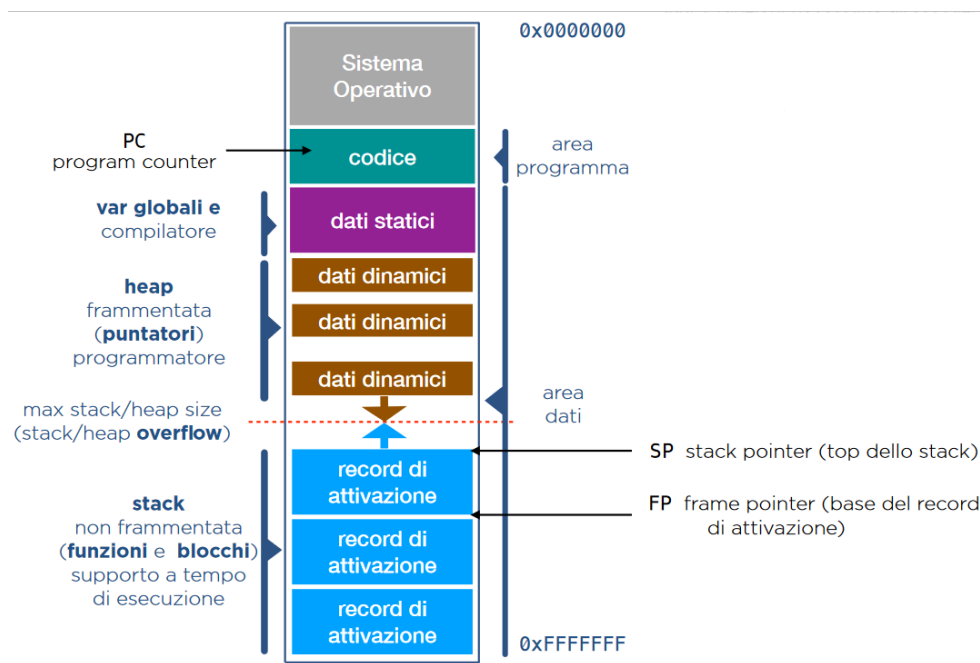


Figure 5: Gestione della memoria di un programma

Note 4.2.1. Partiamo dal presupposto che un **blocco** sia considerato come una funzione senza parametri.

Note 4.2.2. Lo **stack** funziona tramite operazioni di **push** (inserimento di un elemento in cima) e **pop** (rimozione dell'elemento in cima). Può lavorare in due modi:

- **LIFO** (Last In First Out): l'ultimo elemento inserito è il primo ad essere rimosso
- **FIFO** (First In First Out): il primo elemento inserito è il primo ad essere rimosso

4.3 Tipi di ricorsione

Definizione 4.5 (Non lineare). Viene eseguita più di una **chiamata ricorsiva** nel blocco.

Un caso particolare è quando la funzione viene passata come parametro formale della sua stessa definizione e si dice **annidata**.

Definizione 4.6 (Mutua). Quando due o più funzioni sono definite ciascuna in termini dell'altra.

Definizione 4.7 (In coda). La chiamata ricorsiva è l'unica operazione effettuata dalla funzione prima di restituire il controllo alla chiamata. La chiamata in questo caso si definisce **terminale**.

Questa modalità consente di risparmiare spazio di memoria per la gestione dello stack di esecuzione in quando viene gestito come se fosse iterativo e viene creato un solo record di attivazione in più per la gestione dell'indirizzo di ritorno.