

FORMATI

Per “formato” intendiamo proprio il formato per un’applicazione eseguibile.

Nella realtà all’interno di un’applicazione eseguibile noi abbiamo sicuramente le informazioni che permettono poi al sistema operativo, quando l’applicazione va in esercizio, **di capire cosa registrare all’interno** della zona “TEXT”, cosa registrare all’interno di “DATA”/“BSS” e così via.

Tutte queste informazioni vengono comunque ad essere generate quando compiliamo un’applicazione, in modo tale che al lancio dell’applicazione eseguibile, queste vengono generate dal software del sistema operativo per **fare il “setup” del contenuto dell’address space**.

Quindi queste informazioni sono mantenute all’interno di un formato, che è il formato dell’eseguibile.

Su unix, il formato dell’eseguibile che viene generato quando compiliamo con gcc si chiama **ELF** (Executable and Linkable format)

Su sistemi Windows si usa un formato che si chiama PE (Portable Executable), ed è ovviamente il formato dei file .exe.

È compito del tool di compilazione di generare ELF o EXE a partire dai sorgenti del programma e dalle direttive di compilazione

Potrei definire una direttiva di compilazione per cui posso inserire più di una zona nell’address space che funga da TEXT.

-Werror //tutti i warning li trasformi in errori

- 1) ad ogni funzione C compilata e linkata staticamente corrisponderà un’unica routine di istruzioni macchina collocata nella sezione “testo”
- 2) ad ogni variabile globale dichiarata dal programmatore o all’interno di librerie linkate staticamente corrisponderà una locazione (di taglia appropriata) nella sezione “dati”
- 3) la sezione “heap” permette uso di ulteriore memoria, ad esempio tramite la libreria `malloc`
- 4) in tal caso è la stessa libreria `malloc` che tiene traccia di quali aree (buffer) dell’heap sono stati consegnati in uso all’applicazione o rilasciati dall’applicazione
- 5) la sezione “heap” è espandibile fino alla saturazione dell’intero contenitore di memoria dell’applicazione

