

Uso del debugger

Introduzione

Come abbiamo visto nelle scorse lezioni, quando vogliamo eseguire un programma ci basta farlo partire e aspettare che entri in funzione. Potrebbe succedere però che, durante il run time, ci siano degli errori o non ci ritroviamo con i risultati forniti.

Per vederci meglio, entra in gioco il **debugger**, ovvero una speciale modalità di esecuzione in cui possiamo procedere passo a passo nel codice vedendo cosa effettivamente succede. Nel nostro caso, vedremo come si fa il debug sul linguaggio C usando Visual Studio Code, ma il principio è praticamente uguale per tutti i linguaggi (compilati) e gli ambienti di sviluppo.

Prima di cominciare: il concetto di breakpoint

Per eseguire passo passo l'esecuzione del codice sorgente, dobbiamo dire al sistema almeno un punto (possono essere anche di più) dove fermarsi. Se non lo facessimo, il debugger non si fermerebbe e quindi avremmo un'esecuzione classica del programma. Per fermarlo, usiamo il **breakpoint** (letteralmente "punto di rottura") in modo che così il debugger saprà a che riga fermarsi.

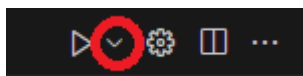
Per applicarlo, basta fare un click destro a fianco al numero della riga in cui vogliamo inserirlo.

```
18  /*Considerando che facciamo partire il count da 0, in questo caso faremo 5 giri nel ciclo*/
19  while (count < 5) {
20      printf("Sono dentro un ciclo while e sono al giro numero %d\n", count);
21      count++; /*FONDAMENTALE aggiornare la variabile che fa da counter, altrimenti avremo un loop infinito*/
22  }
23  printf("\n\n");
```

In questo caso, il debugger si fermerà la prima volta che incontrerà la riga 20.

Come faccio partire il debug?

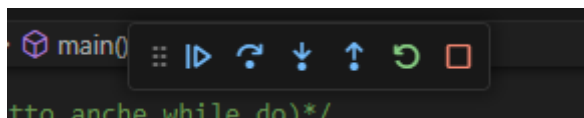
Per far partire il debug, dovrai aprire il file desiderato e cliccare sulla freccetta accanto al triangolo di run (in alto a destra) e selezionare "Debug C/C++ File".



Come si usa il debugger?

Una volta fatto partire, ti troverai nella modalità del debug dove avrai vari strumenti a disposizione.

Innanzitutto, hai gli strumenti per "navigare" nell'esecuzione in alto al centro.



Il **primo pulsante** (oppure premendo F5) ti consente di fare un salto fino al breakpoint successivo. Se non ne trovasse altri, finirebbe l'esecuzione normalmente.

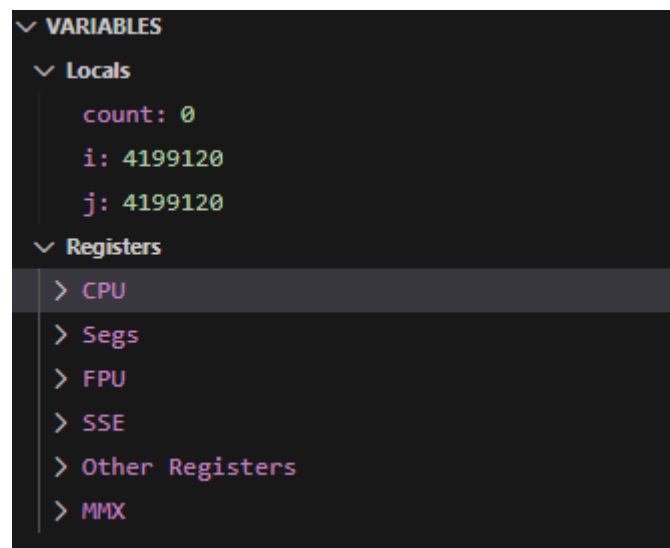
Il **secondo** (oppure premendo F10) ti consente di passare alla riga successiva, indipendentemente dall'istruzione da eseguire. Dunque, se in quella riga c'è una chiamata ad una funzione, non vedremo svolgerla passo a passo ma verrà eseguita "normalmente". Per fare ciò, serve il **terzo pulsante** (oppure premendo F11) che ci farà andare nello specifico anche dentro le funzioni.

Il **quarto** (oppure premendo SHIFT+F11) consente di tornare indietro.

Il **quinto** (oppure premendo CTRL+SHIFT+F5) consente di far ripartire il debug dall'inizio.

L'**ultimo** (oppure premendo SHIFT+F5) interrompe il debug.

A sinistra dello schermo, troviamo vari strumenti, ma quello che ci interessa di più è la sezione variables.



Questa sezione ci da informazioni sul valore delle variabili e dei registri nell'esatto momento in cui stiamo eseguendo.

Possiamo vedere come il count sia a 0 avendo inizializzato, mentre i e j hanno come valore un numero scelto dal compilatore, non essendo ancora state inizializzate.

Se eseguiamo le due istruzioni, potremo vedere che il valore del count si incrementa in base al nostro comando.

