Quinta lezione programmazione 1

Introduzione al linguaggio C

Il linguaggio C nasce come evoluzione del linguaggio B utilizzato per la creazione del sistema operativo Unix. **Dennis Richard** è il creatore del linguaggio C. Si basa sulla programmazione procedurale, modulare e strutturata. Con passare degli anni il C si diffuse sempre di più e pertanto si ebbe una standardizzazione  di questo linguaggio (ANSI) poi ISO  con C11. Seguentemente è stato aggiornato a C18 (versione più recente).

IDE ed editor di testo

I file scritti in linguaggio C (hanno estensione .c).

Un file .p’èc è composto dalla funzione main() che è la funzione principale.

Nelle prime righe del file si dichiarano le librerie (# include <>) che servono per dare una direttiva al preprocessore. Nell’ heder <stdio.h> ci sono i prototipi delle funzioni che si utilizzano per le operazioni di I/O. stdio.h è un file heder che contiene una serie di definizioni come variabili, costanti, prototipi di funzioni e/o procedure (il prototipo è uno stampino della funzione dove si vanno a definire i parametri in ingresso della funzione/procedura) . La funzione restituisce qualcosa la procedura no.

I parametri formali sono quelli che non hanno valore(prototipo o nella funzione stessa),invece, quelli attuali hanno un valore( quando si chiama una funzione nel main()).

Per indicare il “letterale stringa o semplicemente una stringa si usano gli apici  doppi (“”),invece, per indicare un carattere si utilizzano gli apici singoli (‘’). Se, invece, si ha un intero si chiamerà “letterale intero”. Gli /n o /t sono caratteri speciali di tabulazione lo /n manda a capo, invece, lo /t dispone l’output su una riga. I commenti si possono inserire in singola linea o in più linee. (//) (/\* \*/)

Comandi per il terminale

-   **ls**: comando che si utilizza per sapere la lista di file e di directory(cartelle) che si hanno nella directory corrente.

-   Il comando **gcc** serve per eseguire la compilazione del file c vuole come parametri il nome del file(con estensione .c ) da questo si genererà un file eseguibile che per default chiamerà a.out . per l’esecuzione ./a.out

- con – o si rinomina il file eseguibile

Fase di compilazione

1. **Fase di preprocessing:** il preprocessore legge le direttive al preprocesssore (#include). In questo modo diciamo al pre processore che esiste un file header.h  e di includere le librerie nel codice sorgente. Il pre processore legge il file heder lo interpreta (interpretando tutti gli include che ci sono all’interno di stdio.h ricorsivamente) e poi lo concatena con il nostro file sorgente e lo manda al compilatore. (gcc main.c -e con questo comando si va ad eseguire solo la fase di pre-processing)

2.   **Fase di compilazione**:  È la traduzione di tutto il pacchetto prodotto precedentemente in linguaggio macchina. Quando si va a compilare non si deve necessariamente generale eseguibile si può eseguire solamente la compilazione ma non  l’ esecuzione. Se questi file non si eseguono si hanno tante translation  unit che contengono funzioni. una di queste contiene il main e volendo si potrebbero compilare separatamente e si ottengono tanti file Object, file che contengono codice macchine ma non sono file eseguibili si arriva all'eseguibile  del programma mettendo insieme questi file object (linker). Per fare questo da terminale si utilizza il seguente comando: gcc sorgente -o.

     gcc main.c -o main.c nomina il file eseguibile. Per lanciare il programma bisogna scrivere ./maim

Le variabili

Una variabile è uno spazio in memoria adeguato ad ospitare un dato di un certo tipo, è identificata da **un nome**  che è univoco. si dice che il linguaggio C è un linguaggio tipato.Ogni variabile dichiarata nel programma ha un certo indirizzo nella memoria del calcolatore.

int somma=0;

* int(intero) Si sta dichiarando una variabile di tipo  intero quindi in memoria verrà predisposto lo spazio per contenere un intero.
* somma e il nome della variabile o l'identificatore

 Inoltre questa variabile è stata inizializzata con un valore 0.

Quando si dichiara una variabile si sta dicendo al compilatore di predisporre(allocare) uno spazio in memoria per contenere un dato di un certo tipo. invece quando la si Inizializza questa variabile avrà anche un valore. È consigliabile eseguire sempre questa procedura perché le variabili potrebbero avere anche valori sporchi e quindi questo  genererebbe un errore semantico perché  da una determinato operazione  riceviamo un risultato diverso da quello che ci aspettiamo. Si possono dichiarare e inizializzare più variabili nello stesso tempo dello stesso tipo nel modo seguente: int a=2,c=3;

Per mostrare tutti i warning che ha rilevato il compilatore bisogna utilizzare il comando -Wall.

Letterali e variabili

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Tipologie di variabili

* **Int**: intero standard (**va da -2 miliardi a +2 miliardi**)
* **Unsigned int**: intero senza segno (**solo positivo**) quindi i valori che la variabile può contenere diventano 4 miliardi
* **Long**: serve per contenere numeri ancora più grandi di quelli che può contenere un int
* **long long**: serve per contenere numeri ancora più grandi del long
* **float**: numero in virgola mobile in singola precisione (3.5)
* **double**: numero in virgola mobile (ad esempio numeri con più cifre dopo la virgola) es 3.14
* **char**: serve per dichiarare che la variabile in questione dovrà contenere un carattere. Ad esempio  char c=‘c’; (in questo modo sto assegnando un carattere alla variabile c.

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

**operatori di incremento/decremento**

* incremento unitario ++a (PREFISSA)
* decremento unitario --a (PREFISSA)
* decremento unitario a-- (POSTFISSA)
* incremento unitario a++ (POSTFISSA)

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

Per la forma post fissa si vuole assegnare a v il valore di k e poi incrementare k.

Per la forma prefissa prima si incrementa k e poi si assegna a v

Le costanti

Sono dei dati che non cambiano nel tempo o meglio dire in  Run Time.

 Ci sono due modi:

1. **Con la direttiva al preprocessore # define** #define esempio(#define PIGRECO 3.14) all’interno del file header. In fase di compilazione tutte le occorrenze delle costanti utilizzate  vengono sostituite con il valore  letterale.
2. Con la parola chiave **const** questo metodo si utilizza quando il valore non è noto a tempo di compilazione. Per utilizzare questo metodo il valore della costante va inizializzato in fase di creazione. ES const int N=4;

 