

II Esercitazione: S.O. & Fortran 77

Francesco Battista

Corso di Calcolo Numerico

¹DIMA, “Sapienza” University of Rome, Italy

February 9, 2014

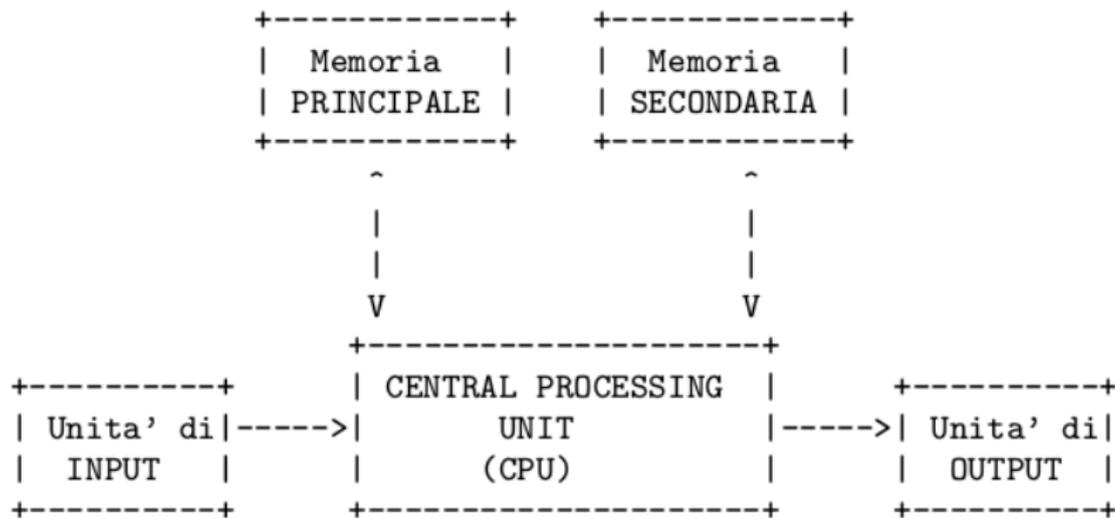
- 1 Brevi cenni sulle architetture informatiche
- 2 Introduzione ad Ubuntu: il terminale
- 3 Introduzione ai linguaggi di programmazione: Fortran 90

Consigli per seguire le esercitazioni

- scrivere i vari programmi autonomamente al computer
- ripetere a casa i programmi che si svolgono in aula
- quando vengono assegnate le esercitazioni svolgerle prima di venire a lezione

Architettura di un calcolatore I

Architettura 'Von Neumann' (1945 circa)



La quantità di informazione è misurata in *Bit*

- Bit (quantità binaria 1/0, acceso/spento)
- 8 bit = Byte

Architettura di un calcolatore II

- *CPU*: frequenza di istruzioni o di *clock* GHz; indirizzamento di memoria 32/64 bit (Intel, Pentium, ...)
- *Memoria principale*: veloce, non persistente e costosa (fino a 4Gb). La più conosciuta è la RAM (Random Access Memory lettura/scrittura)
- *Memoria secondaria*: lenta, persistente e poco costosa. Magnetica Hard Disk (fino a 4Tb). Ottica CD, CD-ROM o DVD.
- *Dispositivi input*: tastiera, mouse, microfono, videocamera...
- *Dispositivi output*: schermo, stampante, altoparlante ...

Fondamenti di Programmazione

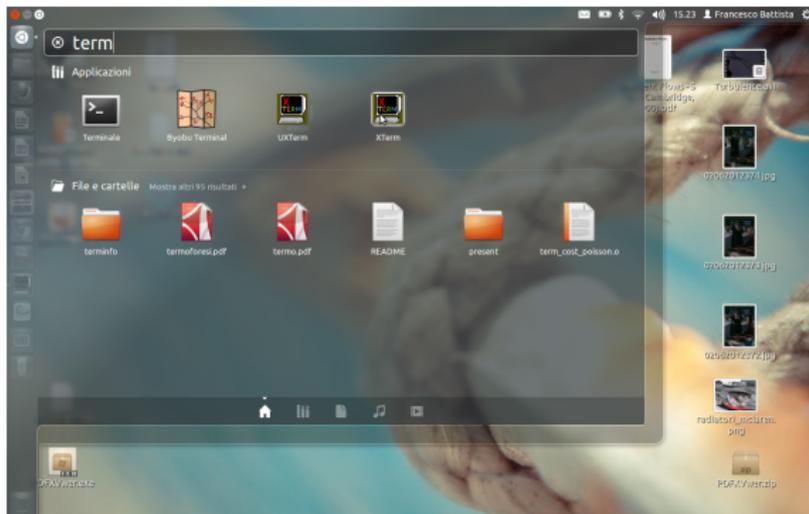
- La *CPU* comprende solo comandi in formato binario.
- Si può ‘dialogare’ con la *CPU* con un linguaggio superiore a quello binario:

Linguaggio	Livello	uso
linguaggio macchina	bassissimo	mai
assembler	basso	solo per elevata efficienza
Fortran, C, C++, Java	alto	maggioranza delle applicazioni
linguaggio naturale	altissimo	ancora non pronto

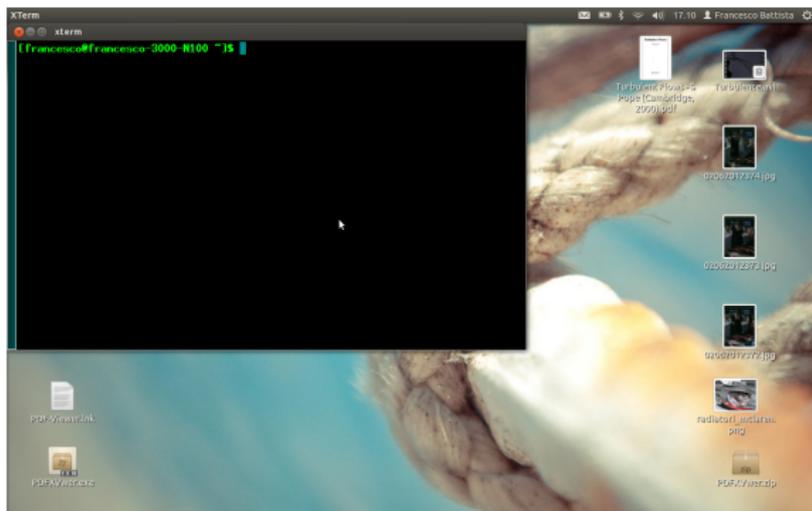
- Esistono diversi programmi commerciali che permettono di scrivere un programma ed eseguirlo
- quello di cui abbiamo bisogno noi è un compilatore ed un editor di testo (gedit, o per i più esperti/temerari vim)

Ottenere il compilatore

- il terminale sarà utile per eseguire i comandi base necessari per compilare ed eseguire un programma (una sorta di prompt del dos di windows).
- barra degli strumenti digitare 'term':



Ottenere il compilatore



- comandi del terminale: riassunto sul sito http://guide.debianizzati.org/index.php/Guida_ai_comandi_da_terminale
- digitare *sudo apt-get install gfortran*
- immettere la password digitata in fase di installazione
- è necessaria la connessione ad internet: farlo a casa

Scrivere un programma: 4 Fasi

- Scrittura:
 - scrittura del programma con apposito *editor*
 - salvataggio su disco
 - file con estensione '.f90'
- Compilazione:
 - trasformazione in linguaggio comprensibile da CPU
 - usiamo compilatore gfortran
 - file con estensione '.o'
- Collegamento:
 - trasformazione in linguaggio comprensibile da CPU
 - usiamo apposito linker generalmente integrato nel compilatore
 - file eseguibile (in Unix assente estensione in Windows .exe)
- Esecuzione/Verifica:
 - si esegue il programma
 - se l'esecuzione è quella attesa l'operazione è conclusa
 - altrimenti si riparte dall'inizio

Perché Fortran?

- il primo linguaggio per cui è stato scritto un compilatore (1954) subendo svariate evoluzioni fino a Fortran 2003
- PRO: molto usato per il calcolo numerico, quindi esistono molte librerie già fatte
- CONTRO: in alcuni casi filosofia un po' vecchia (anni '50)
- linguaggi alternativi: C, C++ e derivati
- la versione 90 non è la più vecchia, quello che faremo funzionerà anche con le versioni più recenti (95, 2003) ed in molti casi sarà varrà anche per la versione 77

Il primo programma

```
1 ! File: ciao.f90
2 ! Primo programma in Fortran 90
3 ! Questo programma stampa un messaggio a schermo
4 PROGRAM ciao
5
6 ! Sezione esecutiva
7 WRITE(*,*) 'Ciao Mondo!'
8
9 ! Sezione conclusiva
10 STOP
11 END
```

- oltre la colonna 71 bisogna andare a capo, mettendo il simbolo & alla fine della riga
- il simbolo '!' indica l'inizio di un commento
- fortran è CASE INSENSITIVE

Il primo programma

- Commenti: sono opzionali ma utili, se non troppi, il compilatore non li vede.
- Intestazione: nome del programma, SIGNIFICATIVO
- Istruzioni
- Conclusione
- Parole Chiave:
PROGRAM WRITE STOP END
- non si possono cambiare per convenzioni sono maiuscole su alcuni editor vengono colorate diversamente.
- il nome del programma
 - ① deve iniziare con un carattere alfabetico
 - ② dal secondo carattere può contenere cifre decimali o '_'
 - ③ può essere lungo al massimo 31 caratteri

Il secondo programma

```
1 ! File: assegnazione.f90
2 ! Questo programma legge e stampa a schermo un numero
3 PROGRAM assegnazione.f90
4
5 ! Sezione dichiarativa
6 IMPLICIT NONE
7 INTEGER :: i
8
9 ! Sezione esecutiva
10 WRITE(*,*) 'Scrivi un numero intero'
11 READ(*,*) i
12
13 WRITE(*,*) 'Hai scritto',i
14
15 ! Sezione conclusiva
16 STOP
17 END
```

Descrizione secondo programma

Sezione dichiarativa

- Prima parte del programma
- nome del programma, dichiarazione di costanti e variabili
- operazioni non eseguibili ma funzionali per il compilatore

Sezione esecutiva

- 'cuore del programma'
- istruzioni eseguibili
- utilizza variabili dichiarate in 'sezione dichiarativa'

Sezione conclusiva

- istruzioni che chiudono il programma

UNA VOLTA SCRITTO IL PROGRAMMA VA
COMPILATO PER OTTENERE L'ESEGUIBILE

```
gfortran nome_file.f90
```

Si ottiene un file che si chiama 'a.out'

```
gfortran -o nome_exe nome_file.f90
```

Si ottiene un file che si chiama 'nome_exe'

l'esecuzione del programma:

'./a.out' oppure './nome_exe'