Usare Python in Linux

Introduzione

28 Febbraio 2014

Outline

1. Laboratorio
   - Struttura e programma delle lezioni
2. Sistema Operativo
   - Definizione
   - Mini tutorial
3. Python
   - Introduzione
   - Programmazione

Outline

1. Laboratorio
   - Struttura e programma delle lezioni
2. Sistema Operativo
   - Definizione
   - Mini tutorial
3. Python
   - Introduzione
   - Programmazione

Lezione tipo

- Correzione degli esercizi svolti autonomamente e consegnati all'indirizzo di posta elettronica: **labinfo.mat.unibo@gmail.com**
- Veloce rassegna di alcuni concetti spiegati a lezione
- Assegnazione di esercizi da svolgere in laboratorio/aula
- Assegnazione di esercizi da svolgere autonomamente

Progetto

- Individuale
- Consegna delle specifiche di progetto circa a metà aprile
- Valutazione:
  - insufficiente $\rightarrow$ si deve ripresentare il progetto a una data di consegna successiva, ma il voto terrà conto della prima consegna
  - sufficiente $\rightarrow$ si assegna un valore compreso tra 0 e 5 da sommare al voto dello scritto

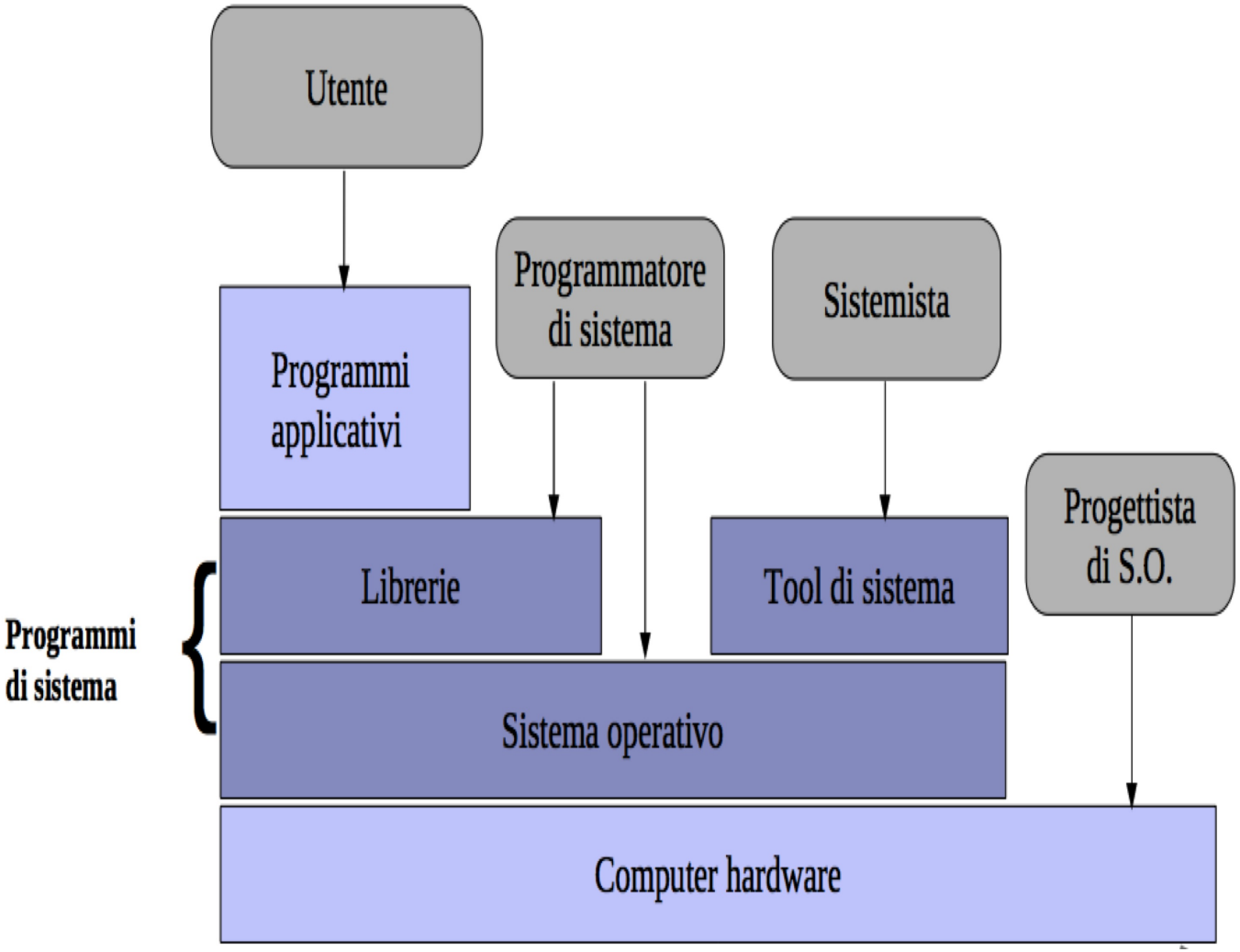
Programma delle lezioni di Laboratorio

- Introduzione a Linux e Python primi costrutti
- Comando di condizione, comando di input, iterazione incondizionata: concetti, esempi ed esercizi
- Sequenze immutabili e iterazione condizionata: concetti, esempi ed esercizi
- Ricorsione: concetto, esempi ed esercizi
- Confronto tra le diverse possibili soluzioni di uno stesso problema: esempi ed esercizi
- Liste: concetti, esempi ed esercizi
- Dizionari: concetti, esempi ed esercizi
- Code pile ed alberi: concetti, esempi ed esercizi
- Esercizi in preparazione all'esame

Outline

1. Laboratorio
   - Struttura e programma delle lezioni
2. Sistema Operativo
   - Definizione
   - Mini tutorial
3. Python
   - Introduzione
   - Programmazione

Sistema Operativo (SO) e interazioni con gli utenti



Definizione di Sistema Operativo

Un Sistema Operativo (SO) è:

- un programma che agisce come intermediario tra Utente, Software Applicativo e Hardware.

Obiettivi e compiti:

- fornire un ambiente nel quale un utente possa eseguire programmi in modo conveniente ed efficiente su un determinato hardware;
- fornire un'interfaccia utente (grafica e non) per poter eseguire applicazioni;
- evitare che i programmi in esecuzione contemporanea entrino in conflitto;
- fornire sevizi ai programmi in esecuzione.

Il S.O. e il suo linguaggio

Il linguaggio del S.O. è rappresentato da tutti i comandi che possono essere riconosciuti dall'interprete del S.O. Ogni comando ha:

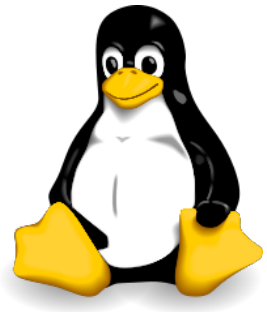
- una sintassi (come devono essere scritti per essere riconosciuti)
- una semantica (quale operazione sarà effettuata)

Comandi

I comandi sono specifici dei vari S.O.:

Comandi Unix	Comandi Dos	Comando
Mkdir	Md	Crea una nuova directory
Rmdir	Rd	Cancella una directory
Ls	Dir	Visualizza i file nella directory
Cp	Copy	Copiare il file
Rm	Del	Elimina file
Man	Help	Aiuto sui comandi
Mv	Move	Muove un file in un'altra posizione

Cos'è Linux?



- È un S.O. Open Source

Ricordiamo che i S.O. moderni hanno tra le altre anche le seguenti funzionalità:

- presentano un'interfaccia (grafica e a linea di comando) per gestire le richieste dell'utente
- gestiscono i vari programmi in esecuzione
- gestiscono le richieste di I/O

Outline

1. Laboratorio
   - Struttura e programma delle lezioni
2. Sistema Operativo
   - Definizione
   - Mini tutorial
3. Python
   - Introduzione
   - Programmazione

Come usare Linux da riga di comando

shell La shell/terminale è l'interfaccia dell'interprete del SO dove un utente inserisce i comandi che vuole eseguire.

comandi Possiamo suddividerli in:

1. Documentali
2. Gestionali:
   - archivi
   - file e directory
   - processi
   - stampa
   - testo
3. Operazionali su file di dati
4. Informativi:
   - sullo spazio su disco
   - su di noi e sul sistema
5. Operazionali di ricerca
6. Vari

Documentali

MAN

sintassi man [section] [option...] NAME.

descrizione man formatta e visualizza le possibili pagine del manuale che riguardano NAME. NAME può essere: un comando, una funzione, un file di configurazione. Se section viene specificata, man cerca la documentazione soltanto in quella sezione. La sezione è la suddivisione in categorie dei comandi

esempio man man $\rightarrow$ entra nel manuale del comando man, per uscire dal manuale usare q

APROPOS

sintassi apropos NAME.

descrizione apropos da informazioni riguardo la voce NAME

esempio apropos man $\rightarrow$ visualizza informazioni sui diversi manuali presenti

Gestione file e directory

LS

sintassi	ls [option...] [NAME_DIRECTORY...] [NAME_FILE...]
descrizione	ls lista il contenuto di ogni directory indicata in [NAME_DIRECTORY...] e i file individuati da [NAME_FILE...].
esempi	ls $\rightarrow$ visualizza tutti i file e le directory presenti nella directory corrente

MKDIR

sintassi	mkdir [option...] NAME_DIRECTORY...
descrizione	mkdir crea le directory specificate con (NOME_DIRECTORY...).
esempio	mkdir /home/nome.cognome/python $\rightarrow$ crea la directory python nella nostra home

Gestione file e directory

RM

sintassi	rm [option...] [NAME_FILE...]
descrizione	rm cancella i file specificati, ma non le directory.
esempio	rm prova.txt → rimuove se esiste il file prova.txt

RMDIR

sintassi	rmdir [option...] NAME_DIRECTORY...
descrizione	rkdir cancella le directory specificate con (NOME_DIRECTORY...).
esempio	rmdir prova → rimuove se esiste la directory prova

Gestione file e directory

CD

sintassi cd [DIRECTORY]

descrizione cd entra nella directory specificata. NB. bisogna descrivere il percorso corretto. Se non è fornita alcuna directory allora entra nella home directory dell'utente.

esempi cd /usr/include → entra nella directory /usr/include

CP

sintassi cp [option...] FILE_ORIGINE... DIRECTORY || cp [option...] FILE_ORIGINE FILE_DESTINAZIONE

descrizione Copia i file specificati con FILE_ORIGINE nella directory DIRECTORY. || Copia il file FILE_ORIGINE nel file FILE_DESTINAZIONE.

esempi cp x.* y.* /home/nome.cognome → copia x,y nella nostra home || cp prova.* prova2.*→copia il file prova in prova2

Lanciare un programma da shell

Per lanciare un programma da shell, indicare il nome del file eseguibile.

- Per esempio apriamo un editor di testo col comando

NANO

- nano nomefile.* $\rightarrow$ si apre l'editor sul file specificato da nomefile.*
- 'E $\rightarrow$ indica la combinazione di tasti Ctrl E

VIM

- un altro esempio di editor è VIM
- VIM opera in due modalità:
  - comando: permette di interagire col programma editor, utilizzando specifiche combinazioni di tasti che corrispondono ai comandi riconosciuti da VIM; come ad esempio: aprire un file, salvare un file, uscire, ricercare e sostituire parole
  - inserimento: permette di scrivere e modificare il testo salvato nel file.

VIM

Passaggio da una modalità all'altra:

- VIM si apre in modalità comando, per cambiare contesto si edita il carattere 'i'
- per tornare alla modalità comando si usa il tasto 'esc'

Comandi principali:

:r nomefile → per aprire un file

:w → per salvare un file senza uscire

:w nomefile → per salvare con nome

:wq → per salvare ed uscire

:q → per uscire (solo se non si sono apportate modifiche)

:q! → per uscire senza salvare

VIM nomefile.* → si apre il file nomefile.* con l'editor VIM

Outline

1. Laboratorio
   - Struttura e programma delle lezioni
2. Sistema Operativo
   - Definizione
   - Mini tutorial
3. Python
   - Introduzione
   - Programmazione

Cos'è Python?

È un linguaggio di programmazione:

- interpretato
- multi-piattaforma
- multi-paradigma.

La versione

- Useremo Python 3.* (il libro di testo è riferito a 2.6 o 2.7).

Python

Per programmare useremo IDLE (Ambiente di sviluppo integrato di Python) Per mandare in esecuzione il IDLE si devono eseguire le seguenti operazioni da terminale:

- accediamo alla directory corretta $\rightarrow$ cd /usr/lib/python3.2/idlelib
- mandiamo in esecuzione il programma digitando $\rightarrow$ python3.2 idle.py
- si apre un editor su cui scriveremo e salveremo i nostri (file) programmi

Outline

1. Laboratorio
   - Struttura e programma delle lezioni
2. Sistema Operativo
   - Definizione
   - Mini tutorial
3. Python
   - Introduzione
   - Programmazione

Comandi

Riprendiamo alcuni comandi:

- `print ('Hello World!')` (scrive Hello World!)
- `x=2` (assegna il valore intero 2 alla variabile x)
- `x=x+5` (al valore di x somma 5 e il risultato lo assegna a x)

Ogni operazione definisce un comando specifico a seconda del tipo di dato che si sta usando.

Tipo di dato

Semplici:

- interi (int) $\rightarrow$ operazioni tra interi
- numeri in virgola mobile (float) $\rightarrow$ operazioni tra float
- numeri complessi (complex) $\rightarrow$ operazioni tra complessi
- valori booleani (bool) $\rightarrow$ operazioni tra booleani

Composto:

- valori stringa (str) $\rightarrow$ operazioni tra stringhe

Operazioni su interi, virgola mobile, complessi

Rappresentazioni:

- interi: M=3
- float: M=3.0
- complessi: M=(3+1j)

Operazioni:

- M+M $\rightarrow$ somma (interi, float, complessi)
- M*M $\rightarrow$ prodotto (interi, float, complessi)
- M/M $\rightarrow$ divisione con risultato intero, float, o complesso
- M%M $\rightarrow$ modulo (solo tra interi)
- M**M $\rightarrow$ elevamento a potenza (interi, float, complessi)

Operazioni stringhe

Rappresentazioni:

- M = 'Prova'; N='casa'

Operazioni:

- M+N $\rightarrow$ concatena la stringa M ed N es. Provacasa
- M*3 $\rightarrow$ concatena 3 volte la stringa contenuta in M es. ProvaProvaProva
- len(M) $\rightarrow$ restituisce la lunghezza di M es. 5
- M[0], $\cdots$ M[len(M)-1] $\rightarrow$ restituisce i singoli caratteri della stringa. es: M[0] $\rightarrow$ P
- M[x:y] $\rightarrow$ restituisce una sotto-striga di quella di partenza partendo dalla posizione x fino alla posizione y-1. es. M[:2] Pr

Esercizi

- Scrivere uno script che assegnati due valori a x e y stampi il perimetro e l'area di un rettangolo
- Scrivere uno script che assegnate due stringhe a x e y stampi la stringa formata dalla concatenazione dai primi caratteri delle due stringhe, dai due caratteri nella posizione $(len(x)//2)$ e dagli ultimi due. (es. 'casa', 'piovra' $\rightarrow$ 'cpsvaa')
- Scrivere uno script che assegnati quattro valori a x1, y1; x2, y2 stampi i valori di m e q della retta passante per i due punti
- Scrivere uno script che assegnata una stringa a x la stampi in ordine inverso.

Cosa abbiamo fatto?

1. Laboratorio
   - Struttura e programma delle lezioni
2. Sistema Operativo
   - Definizione
   - Mini tutorial
3. Python
   - Introduzione
   - Programmazione