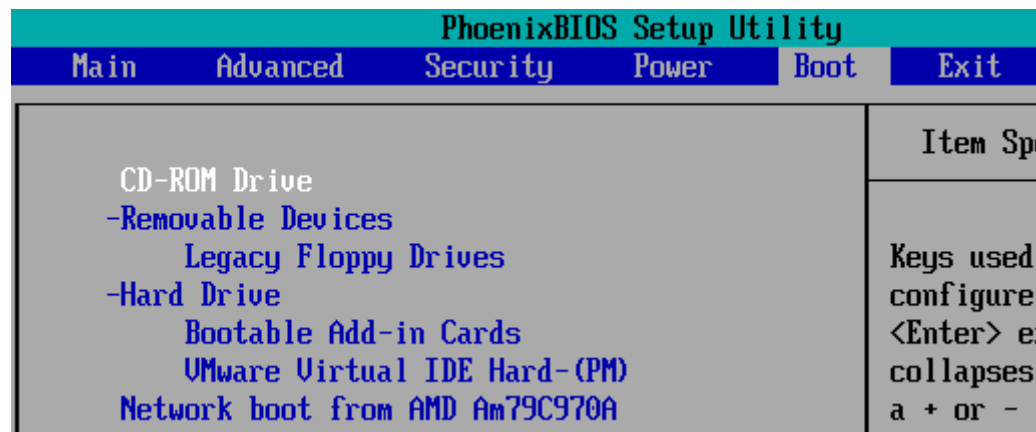


Info
Architettura, JVM
Installazione JDK (Windows, Linux)
Hello world!

Linux: come utilizzarlo

- Avviandolo da distro Live:
 - **download** di un'immagine:
 - ⇒ “Linux live” in Google;
 - **masterizzazione** dell'immagine come bootable CD/DVD con un SW di masterizzazione;
 - **configurazione del BIOS** per il boot da CD/DVD;
 - **boot** da CD/DVD.



Linux: come utilizzarlo

- Installandolo da CD/DVD:
 - **rendendo persistente** una distro Live;
 - **boot da CD/DVD** di installazione;
 - **download** delle immagini .iso:
 - ⇒ **Internet** (cercarne il nome):
 - esempio: “ubuntu download” in Google;
 - ⇒ **riviste** (di solito con CD/DVD allegati);
 - **ATTENZIONE** al **partizionamento**



Terminali e Shell

- **Shell**, strumento capace di:
 - **interpretare** comandi (localizzati tramite PATH);
 - gestire un “ambiente” (variabili d'ambiente);
 - interpretare un vero e proprio **linguaggio di script**;
 - fornire validi strumenti di **interazione con l'utente** e fra comandi (pipe, redirezioni, ...);
 - **svariate shell** diverse (linguaggi diversi):
 - ⇒ /bin/sh -> classica storica;
 - ⇒ /bin/bash -> attualmente la più comune;
 - ⇒ **csh, ksh, tcsh, ...**
- **Terminale**:
 - **GUI** (konsole, xterm, ...) o **testuale** (tty, ...);
 - permette di avviare ed **interagire con una shell**;

Struttura del file-system

- **Struttura logica** del file-system come albero:
 - directory (cartelle) contenenti file e/o directory;
- **Struttura standard** di organizzazione in Linux:
 - “/” rappresenta la **root**, radice dell'albero;
 - “/bin”, “/sbin”, “/usr/bin”, “/usr/sbin” contengono i **principali applicativi installati**;
 - “/etc” contiene i file di configurazione;
 - “/boot” contiene il kernel e l'**immagine di avvio**;
 - “/mnt”, “/media” contengono i riferimenti agli **archivi rimovibili** come CD/DVD, USB drives, ...
 - “/home” contiene le **home degli utenti**;
 - “/usr/share”, “/usr/local/share” contengono la maggior parte delle **applicazioni installate**;

Comandi da shell (breve elenco)

[ls]:	elenco di file e cartelle (locali e non);
[pwd]:	cartella attuale in cui ci troviamo;
[mkdir]:	creazione di una (o più) cartelle;
[rmdir]:	eliminazione di una cartella;
[rm]:	eliminazione file (o directory);
[cd]:	spostamento tra una cartella e l'altra;
[mv]:	spostamento e rinomina di file e cartelle;
[cat]:	output del contenuto di un file;
[less]:	ambiente di lettura di file di testo;
[echo]:	ripete l'input in output;
[grep]:	filtro di line di testo basato su pattern;
[apropos]:	discovery di comandi per topic;
[man]:	manuali dei comandi, cfg-files, libs, ...
[info]:	manualistica più articolata (libri completi).

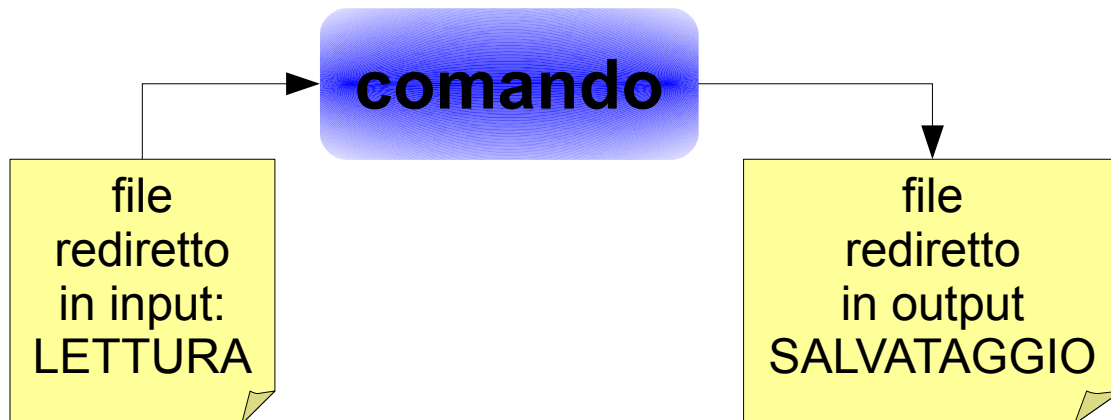
Redirezioni

- **Redirezione dell'input:**

- fornisce input ad un comando da file invece che da tastiera;
 - *comando parametri <file*

- **Redirezione dell'output:**

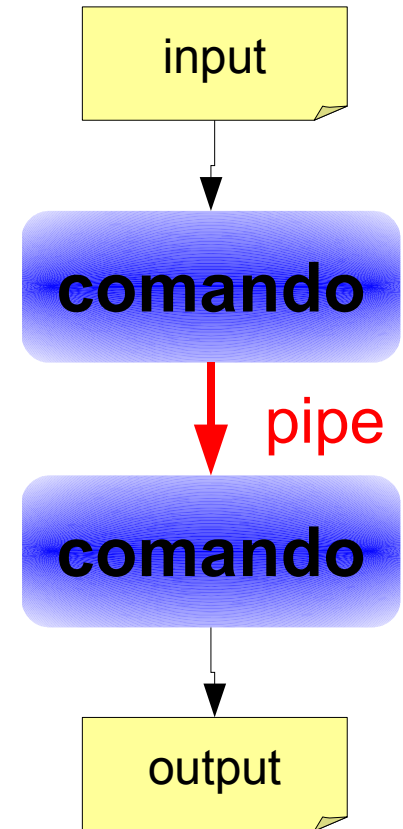
- salva l'output su file invece che mostrarlo nel terminale;
 - *comando parametri >file*
 - *comando parametri >>file*



```
$ ls /usr/bin >~/cmds.txt
$ cd
$ cat cmds.txt
[
411toppm
a2p
...
znew
zsoelim
$ rm cmds.txt
$ touch mieiDati.txt
$ echo $"Dati del gab:\n" \
  >>mieiDati.txt
$ grep ilgab </etc/passwd \
  >>mieiDati.txt
$ cat mieiDati.txt
Dati del gab:
ilgab:x:500:501:Gabriele
Lombardi:/home/ilgab:/bin/bash
```

Pipe e sostituzioni

- **Pipe tra comandi:**
 - consente di “convogliare” l'output di un comando nell'input di un altro: filtering e comandi composti;
 - *comando params | comando params*
- **Sostituzione del risultato:**
 - il risultato di un comando diventa parte del testo di un altro comando (argomenti, nome, percorso);
 - *comando `comando params`*



```
$ df | awk '{print "x+=" $2}END{print "0+x"}' | grep '[0-9]\+' | bc
5629078
$ date +%Y_%m_%d_%H_%M
2008_10_12_17_22
$ mkdir code
$ zip -r code_`date +%Y_%m_%d_%H_%M`.zip code
$ ls
code code_2008_10_12_17_22.zip Desktop
```

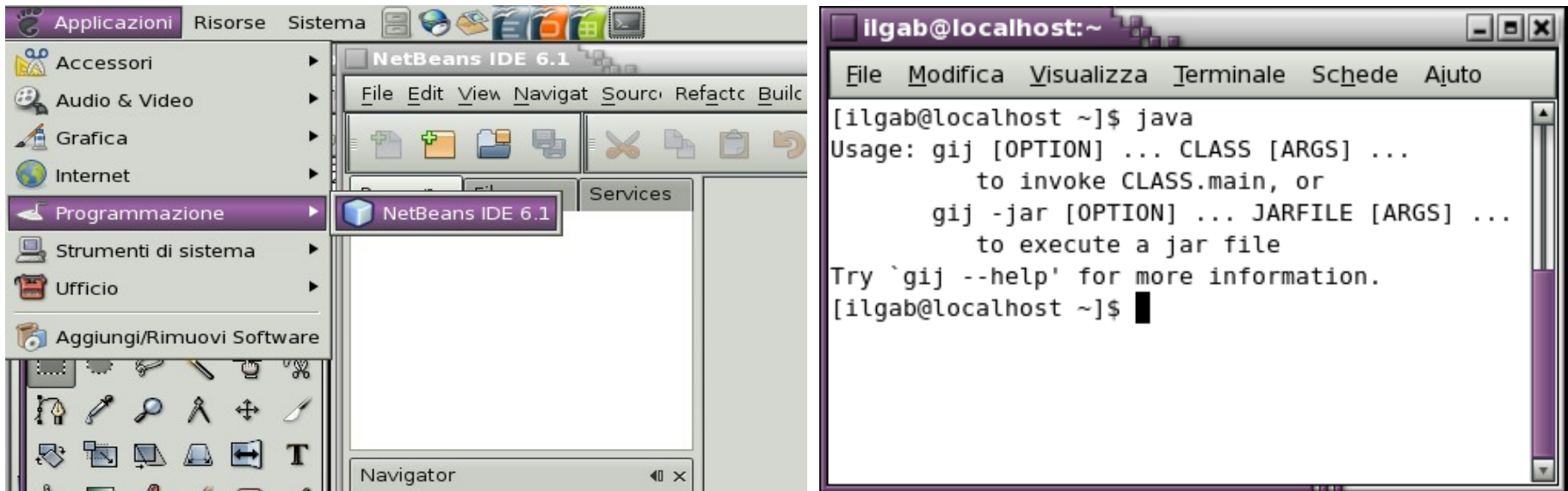
Installazione JDK

- **Download:**
 - <http://java.sun.com/javase/>
 - scegliere l'ultima versione...
 - ...si scelga la piattaforma Linux;
- **Installazione:**
 - copiare nella propria home il file di installazione del JDK;
 - eseguirlo come “./nomeInstaller”:
 - ⇒ `./jdk-6u6-nb-6_1-linux.sh`
 - si avvia l'installazione guidata...
 - ⇒ ...come utente privilegiato root!!!
- **Contenuti:**
 - JDK con compilatore, JVM e altro;
 - IDE NetBeans.



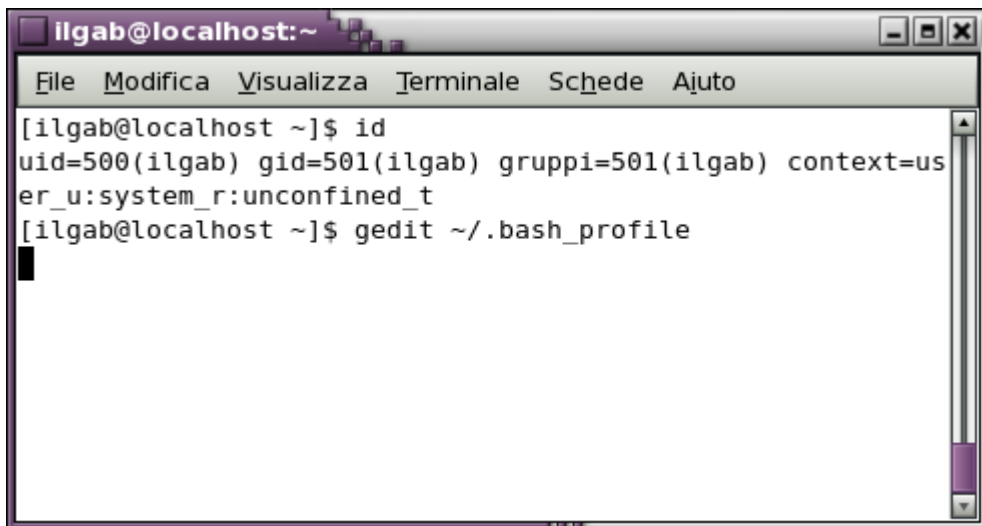
Procedura di installazione

- Selezione della **cartella di destinazione**:
 - `/usr/local/share/java/jdk1.6.0_06`
 - `/usr/local/share/netbeans-6.1`
- Con queste poche info. l'installazione si avvia!!!
- Dal menu è ora disponibile:
 - **Programmazione -> NetBeans IDE 6.1**
- Da riga di comando:
 - **javac** (compilatore), **java** (interprete), ...

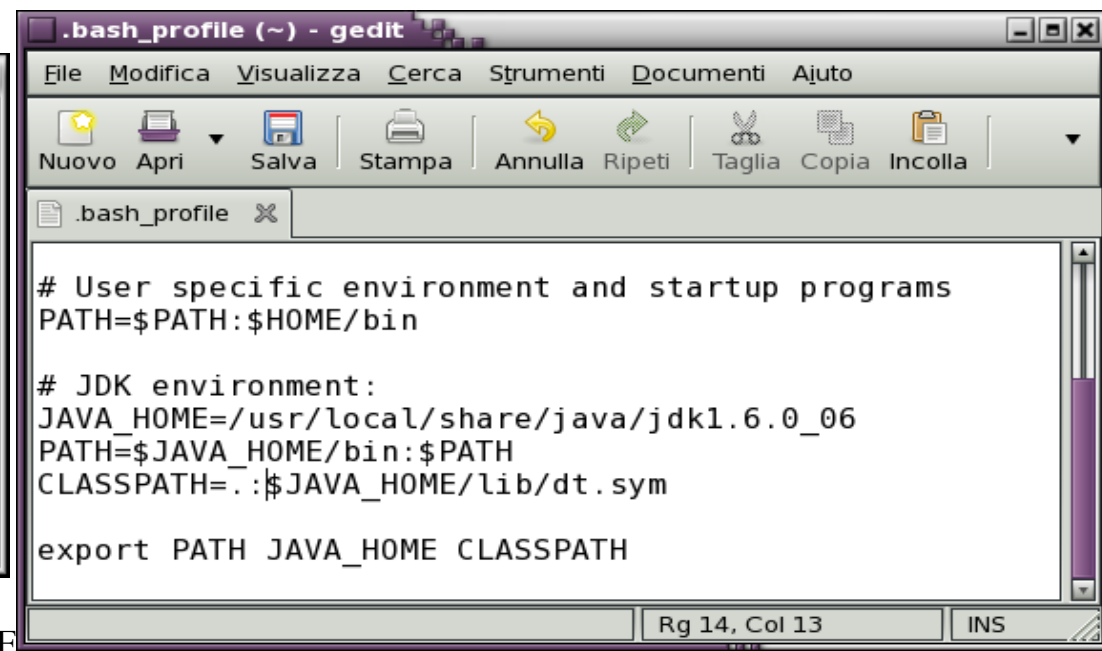


Variabili d'ambiente

- Definiscono delle informazioni che il SO usa;
 - utente, home, lingua, ...
 - ...elenco (separato da “:”) dei **percorsi di ricerca** dei comandi, home del JDK, **percorso delle classi standard** (e non) a cui accede il JDK;
 - ogni utente può definire le proprie variabili:
 - ⇒ nel file `~/.bash_profile`... editabile con gedit.



```
ilgab@localhost:~  
File Modifica Visualizza Terminale Schede Ajuto  
[ilgab@localhost ~]$ id  
uid=500(ilgab) gid=501(ilgab) gruppi=501(ilgab) context=us  
er_u:system_r:unconfined_t  
[ilgab@localhost ~]$ gedit ~/.bash_profile
```



```
.bash_profile (~) - gedit  
File Modifica Visualizza Cerca Strumenti Documenti Ajuto  
Nuovo Apri Salva Stampa Annulla Ripeti Taglia Copia Incolla  
.bash_profile x  
# User specific environment and startup programs  
PATH=$PATH:$HOME/bin  
  
# JDK environment:  
JAVA_HOME=/usr/local/share/java/jdk1.6.0_06  
PATH=$JAVA_HOME/bin:$PATH  
CLASSPATH=.:$JAVA_HOME/lib/dt.sym  
  
export PATH JAVA_HOME CLASSPATH  
Rg 14, Col 13 INS
```

Primo esempio

- Solo questa volta per dichiarare le variabili:
 - `source ~/.bash_profile`
- Editare un file di esempio nella cartella “code”:
 - `cd ~/code`
 - `gedit CiaoMondo.java`
- Compilare ed eseguire:
 - `javac *.java`
 - `java CiaoMondo`

```
public class CiaoMondo {  
    public static void main(String argv[]) {  
        System.out.println("Ciao mondo!");  
    }  
}
```

```
$ javac *.java  
$ java CiaoMondo  
CiaoMondo!  
$ ls  
CiaoMondo.java  
CiaoMondo.class
```

Strumenti di debugging

- “**A manina**”:
 - visualizzazione di messaggi di debug su stderr;
 - realizzazione di strumenti di logging;
 - utilizzo di librerie (**Junit** per il testing, le **API standard di J2SE** per il logging);
- Utilizzando gli **strumenti forniti** con il JDK:
 - potente debugger da linea di comando: **jdb**
 - ⇒ **jdb CiaoMondo**
 - ⇒ **stop in CiaoMondo.main**
 - ⇒ **run**
 - ⇒ **step**
 - ⇒ **step**