

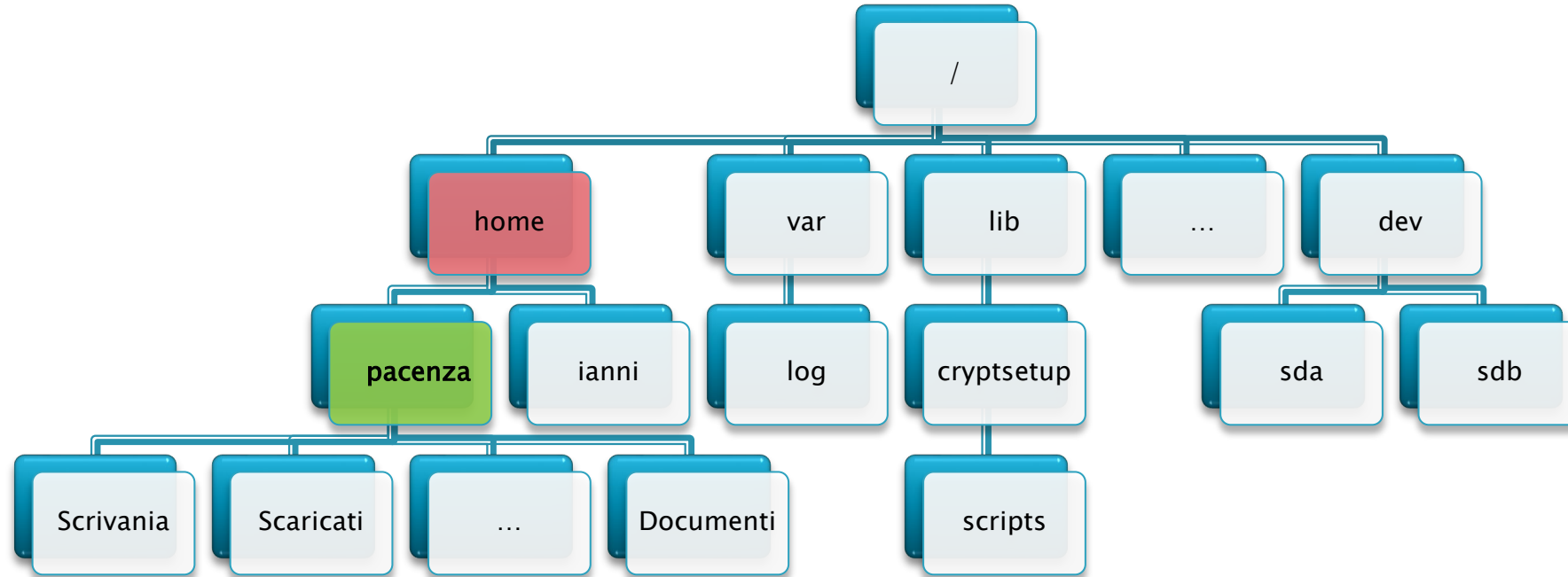
Command Line Fu



La shell Linux

- ▶ Come si accede a BASH
 - Tasti utili: Ctrl-Alt-F1, Ctrl-Alt-F7
- ▶ Il Primo comando : **exit** (CASE SENSITIVE!!)
 - Si può uscire con CTRL-D
- ▶ Il secondo comando: **ls**
 - **Opzioni:** `ls -l`, `ls -a`, `ls -R`
 - `ls --full-time`
 - **Parametri:** `ls /usr`
- ▶ Come avere aiuto: **man**, **info**, opzione **--help**

Path assoluti e relativi



Directory corrente	Directory destinazione	Path assoluto	Path relativo
pacenza	Scrivania	/home/pacenza/Scrivania	./Scrivania
pacenza	cryptsetup	/lib/cryptsetup	../../lib/cryptsetup
pacenza	log	/var/log	../../var/log

Percorsi assoluti nei vostri programmi == Problemi

Le directories

- ▶ Qui si usa “/” anzichè “\”
- ▶ I riferimenti alle directory possono essere
 - Assoluti: `/usr/bin`
 - Relativi alla **directory corrente**: `../usr`
- ▶ Simboli speciali: `.` e `..`
- ▶ Alcune directory predefinite

Directory	Contenuto
<code>/</code>	Root directory
<code>/bin, /sbin, /usr/bin, /usr/sbin</code>	Eseguibili
<code>/dev</code>	Dispositivi (tutto è un file!)
<code>/etc</code>	File di configurazione
<code>/home</code>	Home degli utenti
<code>/lib</code>	Librerie
<code>/mnt, /media</code>	File system esterni (non ci sono lettere di drive)
<code>/opt</code>	Componenti opzionali
<code>/tmp</code>	File temporanei
<code>/usr</code>	Molti eseguibili e tanto altro
<code>/var</code>	File variabili, soggetti a continue modifiche (es. file di log)
<code>/proc</code>	Processi e varie

Come navigare le directory

In ogni momento la shell ha una sua *cartella corrente*.
Ogni utente possiede una sua cartella `$HOME`

Comando	Descrizione
<code>mkdir <nomedir></code>	Crea una nuova cartella dal nome <code><nomedir></code>
<code>cd <nomedir></code>	Ci si sposta nella cartella <code><nomedir></code>
<code>cd</code>	Ci si sposta in <code>\$HOME</code>
<code>rmdir <nomedir></code>	Rimuove la cartella dal nome <code><nomedir></code> ma SOLO se è vuota
<code>rm -rf <nomedir></code>	Rimuove la cartella dal nome <code><nomedir></code> e ricorsivamente tutto il suo contenuto***
<code>pwd</code>	Mostra il path in cui ci si trova
<code>ls <nomedir></code>	Lista il contenuto della cartella <code><nomedir></code>

*** È PERICOLOSO. Tutto ciò che è stato eliminato non potrà più essere recuperato

Come gestire i file

Comando	Descrizione
cp file1 file2	Crea una copia del file1 in file2
cp file1 .. filex directory	Crea una copia di tutti i file specificati in directory
mv file1 file2	Rinomina il file1 in file2
mv file1 .. filex directory	Sposta tutti i file specificati nella in directory
rm file1	Rimuove il file1
less file1	Legge il contenuto del file1 in modalità interattiva
cat file1	Stampa su terminale il contenuto di file1
nano file1	Modifica il file1 da terminale (alias: pico)
file filename	Analizza il tipo di file di filename
touch file1	Crea un nuovo file vuoto dal nome file1

Ricerca tra directory e file

Comando	Descrizione
find <nomedir> -name file1	Cerca il file di nome file1 a partire dalla directory <nomedir>
find <nomedir> -name *.txt	Cerca i file che hanno estensione .txt a partire dalla directory <nomedir>
grep parola <nomedir>	Cerca in tutti i file collocati nella directory <nomedir> la parola parola
grep parola file1	Mostra solo le righe del file1 che contengono la parola parola



Caratteri Jolly

▶ Asterisco `"*"`

- Sta per 0 o più caratteri
- Esempi:
 - `*.txt`
 - `*txt`
 - `*txt*`

▶ Punto interrogativo `"?"`

- Sta per un qualsiasi singolo carattere
- Esempi: `x??`, `t?t`

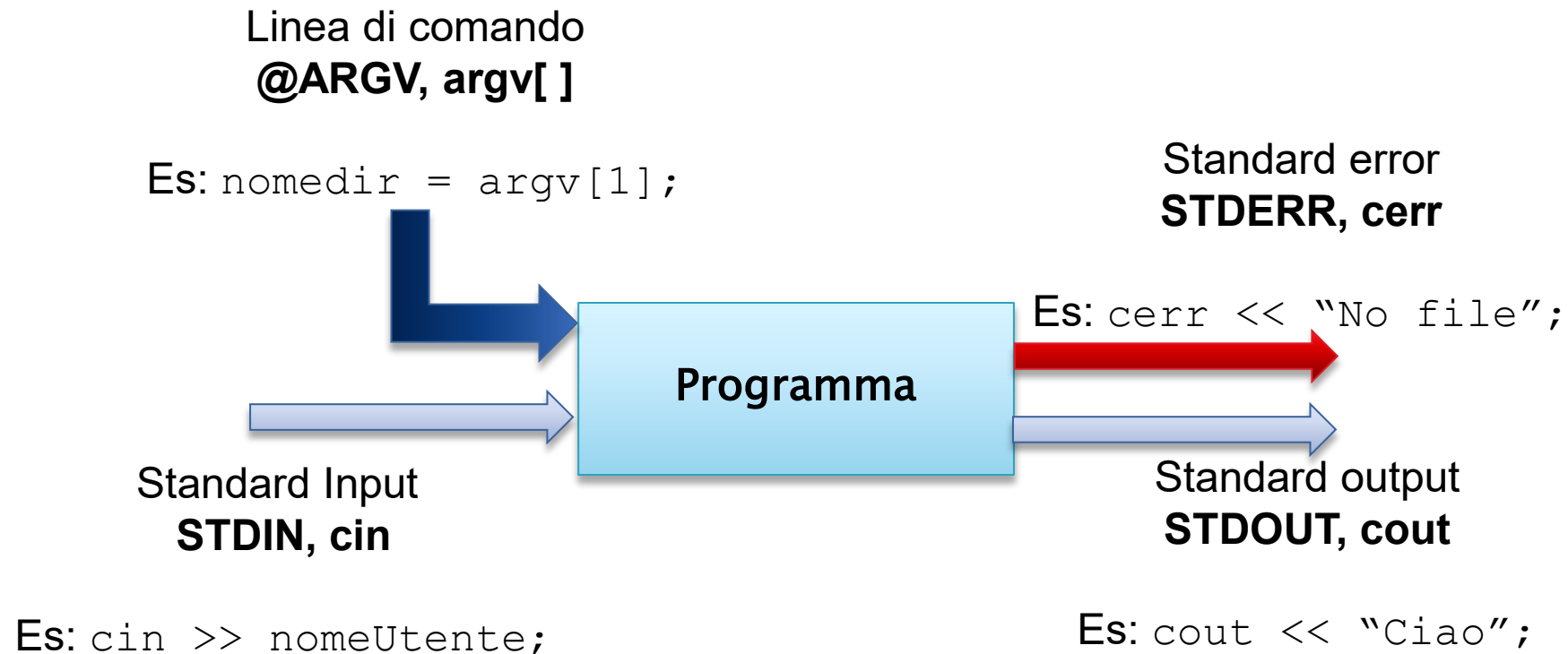
▶ Parentesi quadre `[]` e `"!"` oppure `"^"`

- `[AB]*` tutti i file che cominciano per A o B

Navigare tra i comandi e shortcut

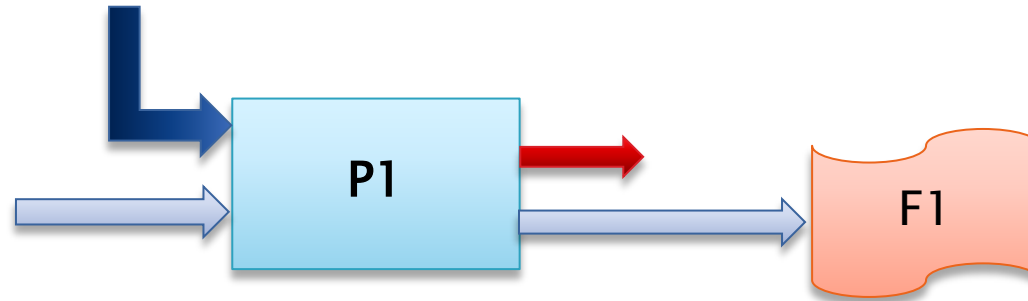
Comando	Descrizione
CTRL + R	Avvia la ricerca di comandi tra la <code>history</code> dei comandi eseguiti in passato
Tasti freccia	Scorre i precedenti comandi
Tab	Completa i nomi di file e dei comandi
CTRL + L	Ripulisce il terminale (scorciatoia comando <code>clear</code>)
CTRL + C	Interrompe l'esecuzione di un comando

Programmi console e loro canali di input

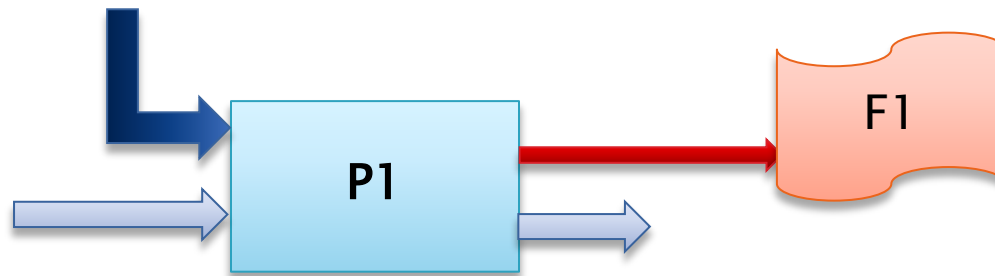


- La linea di comando viene specificata **PRIMA** che il programma sia lanciato (argv)
- I valori vengono letti da STDIN **MENTRE** il programma gira

Redirezione stdout ed stderr

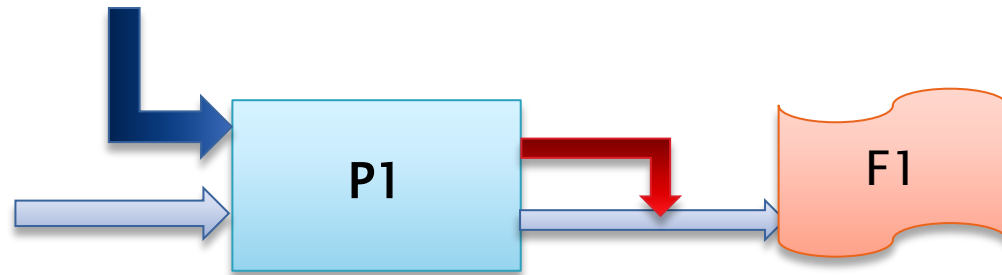


- `P1 > F1` STDOUT di P1 finisce sul *File* F1



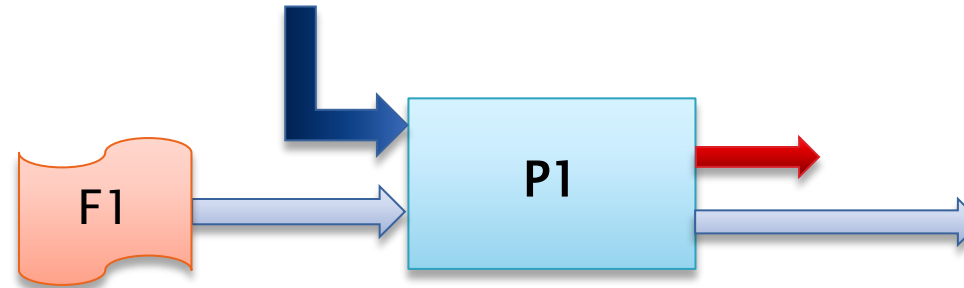
- `P1 2> F1` STDERR di P1 finisce sul *File* F1

Redirezione stdout ed stderr



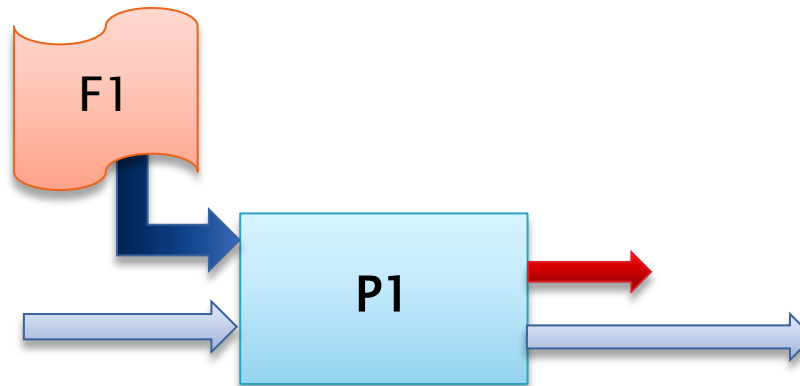
- **P1 >& F1** STDERR e STDOUT di P1 finiscono entrambi sul *File* F1

Redirezione e piping



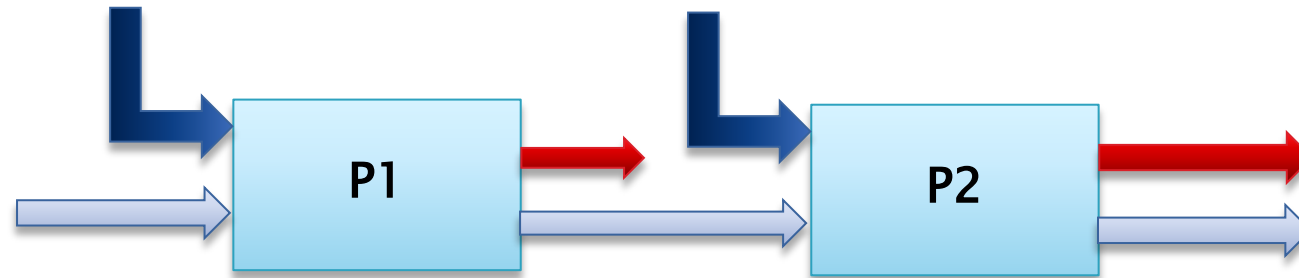
- **P1 < F1** STDIN di P1 riceve IL CONTENUTO del *File* F1

NON E' LA STESSA COSA DI



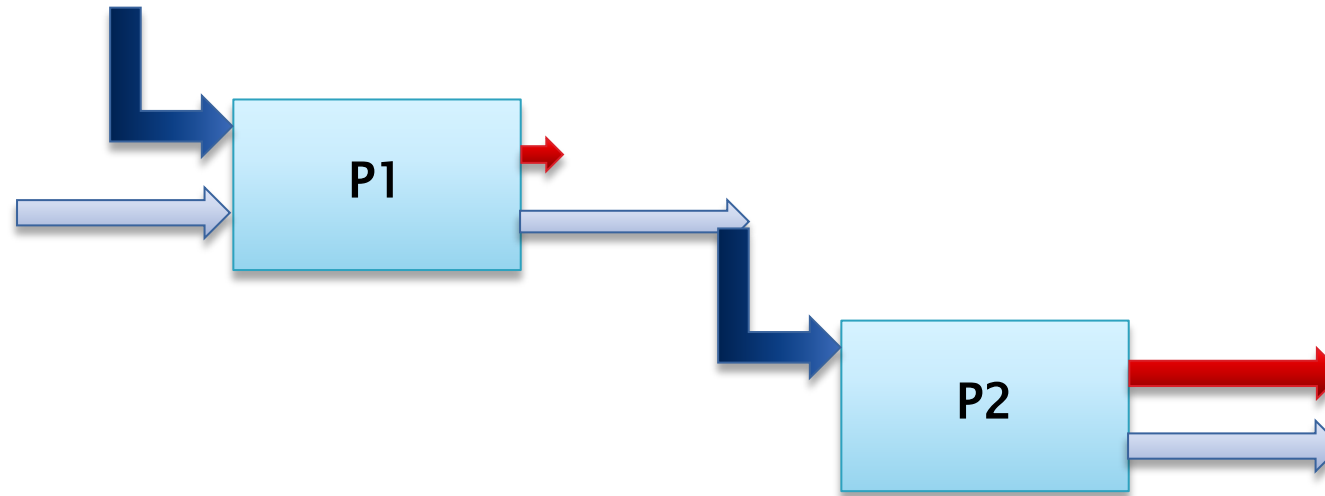
- **P1 F1** \rightarrow ARGV[0] = "F1";

Redirezione e piping



- **P1 | P2** Esegue P1 e P2: STDOUT di P1 diventa STDIN di P2

Redirezione e piping: xargs



- **P1 | xargs P2** Esegue P1 e P2: STDOUT di P1 diventa ARGV di P2



Redirezione e piping

- ▶ Non dimenticate che un programma è sempre agganciato a un canale di input (cin) e alcuni di output (cout e cerr)
- ▶ Agganciare **cout** a qualcosa di diverso dallo schermo:
 - `ls > out.txt` (crea un file out.txt)
 - `ls >> out.txt` (appende a out.txt)
- ▶ Agganciare **cin** a qualcosa di diverso
 - `grep "cat" < out.txt`
- ▶ Usare le pipe (leggi 'PAIP')
 - `cat /etc/services | sort`
- ▶ **Si possono combinare più pipe**
 - `cat /etc/services | sort | tail -n 50 | less`

Variabili di ambiente

- ▶ **env** Mostra tutte queste variabili
- ▶ **export** Cambia il valore di una di esse

- ▶ Variabili speciali
 - \$SHELL, \$PATH, \$HOME

Gestione processi e job

Comando	Descrizione
<code>ps</code>	Mostra la lista dei processi attivi e i loro id
<code>top</code>	Mostra la lista dei processi attivi <i>in tempo reale</i> dei processi e le attività gestite dal kernel
<code>htop</code>	Come <code>top</code> ma semplificato e con possibilità di interazione da parte dell'utente
<code>kill <pid></code>	Termina il processo con il process id (pid) specificato
<code>jobs</code>	Mostra i task eseguiti in background dall'utente
<code>fg</code>	Sposta un task da background in foreground
<code>bg</code>	Riprende l'esecuzione di un task lasciato in sospeso in background

Diritti di accesso

- ▶ chmod → tutorial semplificato
 - <https://catcode.com/teachmod/>

'r' e 'x' per cartelle

Permission	Can Do	Can't Do
r	- List folder contents (view file and folder names)	- Access (enter) the folder
		- Read or write files inside the folder
x	- Access (enter) the folder	- List folder contents (view file and folder names)
	- Read or write files inside the folder (if permitted)	