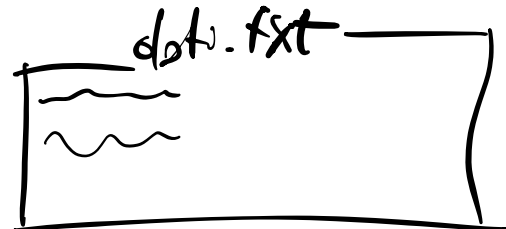


I/O CON FILE

→ ./pippo
Ciao come stai?
Bene e tu?
Ho un po' di raffreddore
Mi piace, curati...

REDIREZIONE DI STDIN

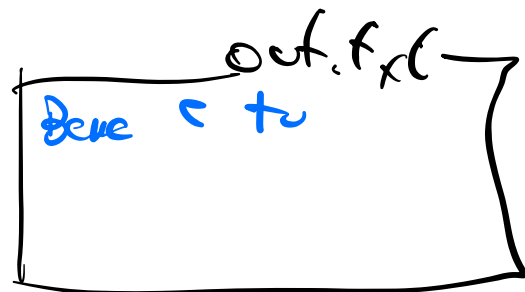
→ ./pippo < dati.txt



REDIREZIONE DI STDOUT

→ ./pippo > out.txt

Ciao come stai?
Ho un po' di raffreddore



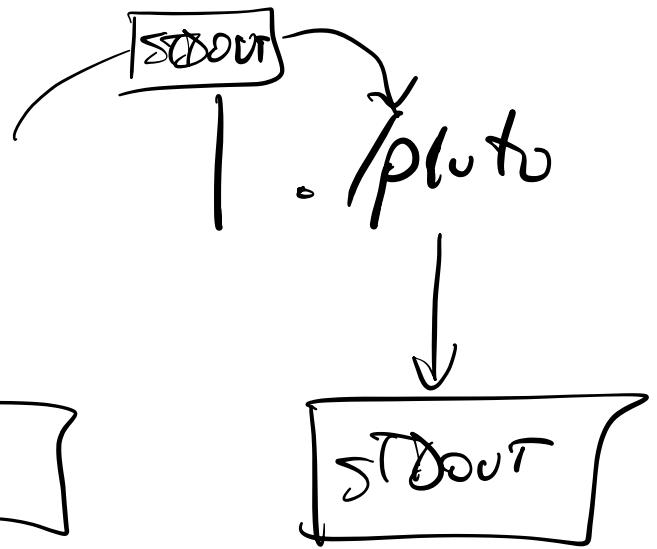
REDIREZIONE DI I/O

→ ./pippo < dati.txt > out.txt

PIPING

~>

./pippo



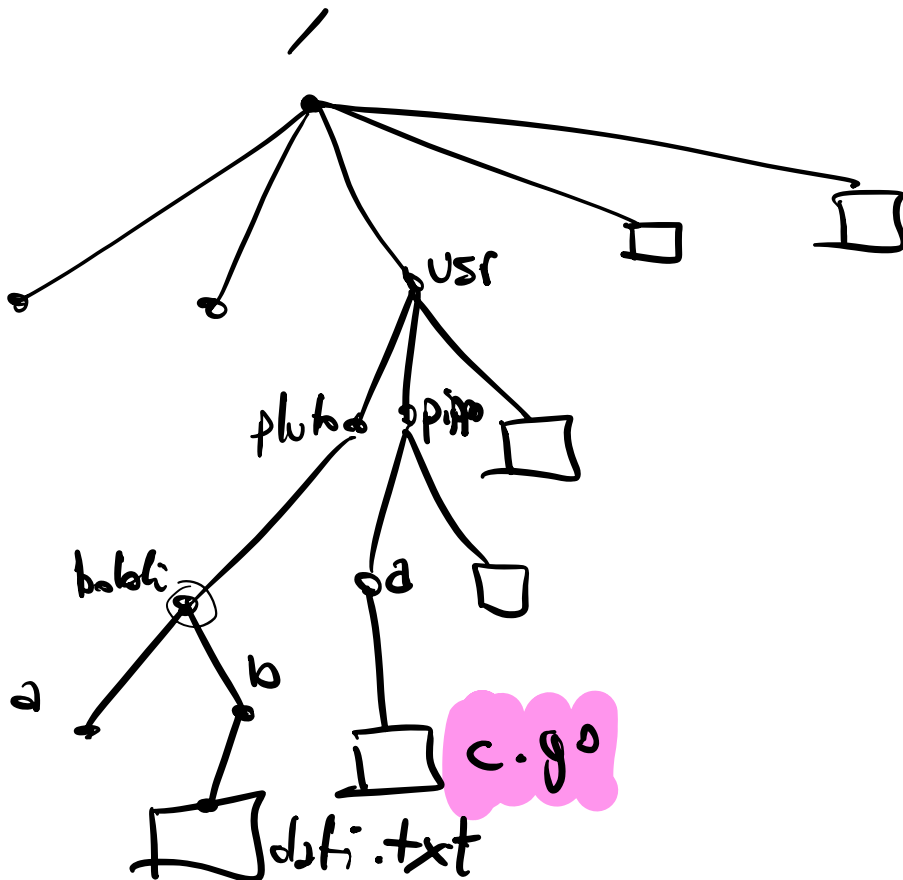
FUNZIONI DI APERTURA

LETTURA

os.Open(string) → (File*, error |

SCRITTURA

os.Create(string) → (File*, error)



```
/usr/pippo/a/c.go  
c:\usr\pippo\ a\c.go  
b/dati.txt  
.. / .. /pippo/a/c.go
```

CHIUSURA

f.Close()

```
f, err := os.Create("pippo.txt")  
....
```

```
....  


f.Close()


```

con defer

```
f, err := os.Create("pippo.txt")  
defer f.Close()
```

LETTURA "RAW"

f. Read (b []byte) (int, err)
 ↓ ↓
 len (b) #byte letti
 0, io.EOF

LETTURE NON - "RAW"

Qualunque si possa usare
un io.Reader (e.g.
in bufio.NewScanner)
potete usare f

SCRITTURA "RAW"

f. Write (b []byte) (int, error)
 ↓ ↓
 DA SCRIVERE #byte di scr.
 nil

SCRITTURE NON - "RAW"

Qualunque si possa usare
un io.Writer (e.g.
in fmt.Fprintf)
potete usare f

potere usare v