

# SSH e script

prova ad eseguire il comando

**ssh localhost.**

Che cosa accade? ti pare un comportamento accettabile?

# ssh-keygen

ssh consente di accettare come credenziali di login la chiave pubblica di un utente. Prova a generare una coppia di chiavi.

**ssh-keygen -b 1024 -t dsa**

Quanti bit



Tipo  
(rsa, dsa)

# ssh-key playback

```
$ ssh-keygen -b 1024 -t dsa
```

Generating public/private dsa key pair.

**Enter file in which to save the key (/home/ghedo/.ssh/id\_dsa):**

**Enter passphrase (empty for no passphrase):**

**Enter same passphrase again:**

Your identification has been saved in /home/ghedo/.ssh/id\_dsa.

Your public key has been saved in /home/ghedo/.ssh/id\_dsa.pub.

The key fingerprint is:

```
80:4c:86:b3:13:87:80:a2:ad:f2:d9:f9:b7:0d:72:ed
```

```
ghedo@muvo.ing.unibo.it
```

# .ssh -> explore

Come sono i diritti dei file dentro .ssh? sono ragionevoli? e quelli di .ssh? Hai notato le differenze fra i vari file?

se il demone ssh trova un file **authorized\_keys2** contenente una chiave pubblica di un utente (id\_dsa.pub) considererà l'utente proprietario della chiave sicuro e lo farà entrare **senza** chiedere la password!

# ssh:prova tu!

Prova a predisporre le cose si che eseguendo

**ssh localhost**

non ti venga richiesta la password

puoi far si che un tuo collega possa entrare senza chiederti la password?

# ssh con argomenti

```
ssh [<user>@]<server> [<comando>[<argomenti>]]
```

Esempi

```
ssh localhost ps  
ssh localhost cat /etc/passwd  
ssh localhost ls
```

**Prima pensa:** che cosa otterrai come output  
dalla sequenza di questi due comandi?

```
cd .ssh
```

```
ssh localhost ls *
```

# ssh e stelle

**Prima pensa:** che cosa otterrai come output dalla sequenza di questi due comandi?

```
cd .ssh
```

```
ssh localhost 'ls *'
```

**Prima pensa:** che cosa otterrai come output dai due comandi (**partendo sempre dalla home -> cd**)?

```
ssh localhost 'cd .ssh ; cat *.pub > ghd'
```

```
ssh localhost 'cd .ssh ; cat *.pub' > ghd
```

# ssh e pipe

**Prima pensa:** che cosa otterrai come output dai comandi (**partendo sempre dalla home -> cd**)?

```
ls id* | ssh localhost 'cd .ssh ; wc -l' | tee ghd
```

```
ls id* | ssh localhost 'cd .ssh ; wc -l | tee ghd'
```



# eempio...

```
ssh host1 'mkisofs -l -f /song' | ssh host2 'cdrecord -dev=4,0 -'
```

# Compitini

1. fare uno script che restituisca il numero di processi in esecuzione su una arbitraria macchina
2. fare uno script che data una lista di macchine dica quella che ha meno processi che girano
3. fare uno script che dato un file di testo composto da 1 parola per ogni riga lo faccia remotamente ordinare alla macchina che ha meno processi in esecuzione