

Linux dalla console

PARTE 2: SCRIPTING, VARIABILI, DATI, FILE



Agenda

- ▶ Esecuzione di comandi
- ▶ Cos'è uno script
- ▶ Variabili
- ▶ Controllo di flusso
- ▶ Cicli
- ▶ Subshell
- ▶ Input utente
- ▶ Argomenti
- ▶ I/O su file

Esecuzione di comandi

```
$ bash -c '<comando 1>; <comando 2>; \
```

```
$ <comando 3>'
```

Eseguiamo più comandi

```
$ <comando 1> && <comando 2> | | <comando 3>
```

```
$ <comando con tempi esecuzione lunghi> &
```

```
$ fg
```

```
$ nohup <comando> >logfile
```

Cos'è uno script?

- ▶ File di testo
- ▶ Contiene istruzioni
 - ▶ Eseguite nell'ordine in cui sono scritte
- ▶ In bash esistono comandi **interni** ed **esterni**
 - ▶ **Interni** sono integrati in bash (if, for, test, while, ...)
 - ▶ **Esterni** sono applicazioni nella path (ls, mkdir, grep, awk, true, ...)



Esecuzione di uno script

Esecuzione di uno script

```
$ bash script.sh
```

Esecuzione di uno script

```
$ chmod +x script.sh
```

```
$ ./script.sh
```


Esecuzione di uno script

```
$ script.sh
```

```
script.sh: command not found
```

```
$ ./script.sh
```

```
-bash: ./script.sh: Permission denied
```

Privilegi di esecuzione

```
$ chmod +x nomefile.sh
```

Lo script può essere eseguito da tutti

```
$ chmod u+x nomefile.sh
```

Lo script può essere eseguito solo dal proprietario

Esecuzione dello script ovunque

```
$ echo echo ok > script.sh
$ chmod +x script.sh
$ cp script.sh $HOME/bin/
$ cd /tmp
$ script.sh
Ok
```

```
$ echo echo ok > script.sh
$ chmod +x script.sh
$ cd /tmp
$ script.sh
script.sh: command not found
```

Scripting con bash

Un esempio di script

```
#!/bin/bash
```

```
#commento
```

```
<comando>
```

```
<comando>
```

```
<comando>
```

```
<comando>
```

```
<comando>
```

```
...
```

Impostazione di una variabile

```
variabile="42"
```

```
# Impostazione variabile d'ambiente
```

```
export EDITOR="vim"
```

Cancellazione di una variabile

```
$ unset variabile
```

Uso della variabile

```
$ echo $variabile
```

```
42
```

```
$ echo ${variabile}
```

```
42
```


Controllo di flusso con if

```
If [[ ... ]]; then  
    <comando>  
elif [[ ... ]]; then  
    <comando>  
else  
    <comando>  
fi
```

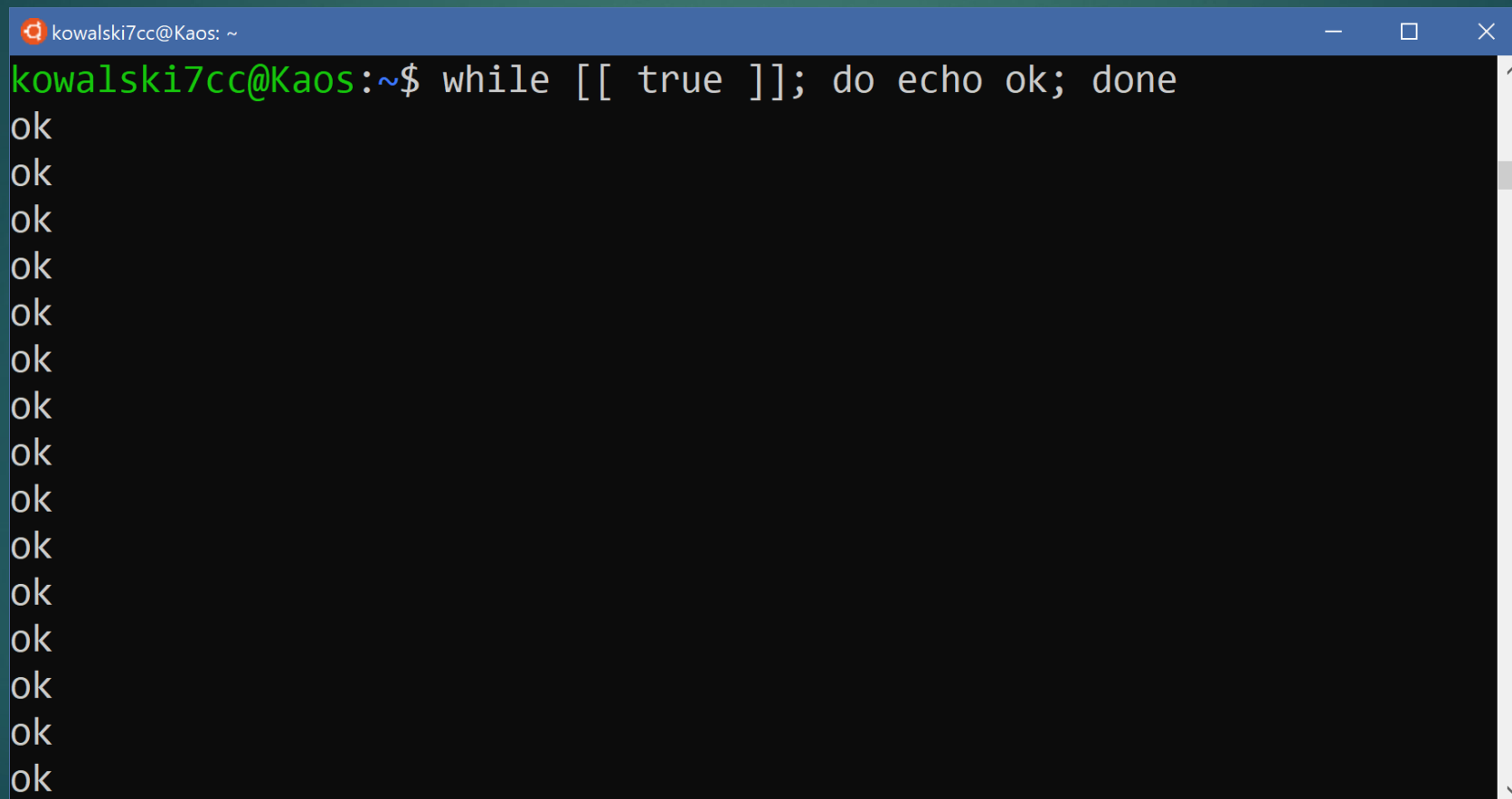
Controllo di flusso con case

```
case $var in
  1)
    <comando 1>
    ;;
  2-5)
    <comando 2>
    ;;
  *)
    <comando default>
    ;;
esac
```

Ciclo while

```
while [[ ... ]]; do  
    <comando>  
    ...  
done
```

Ciclo while

A terminal window with a blue title bar containing the text 'kowalski7cc@Kaos: ~' and standard window controls. The terminal has a black background with green text for the prompt and white text for the command and output. The command 'while [[true]]; do echo ok; done' is entered at the prompt. Below the command, the word 'ok' is printed repeatedly on multiple lines, demonstrating an infinite loop. A vertical scrollbar is visible on the right side of the terminal window.

```
kowalski7cc@Kaos: ~$ while [[ true ]]; do echo ok; done
ok
ok
ok
ok
ok
ok
ok
ok
ok
ok
ok
ok
ok
ok
ok
ok
```

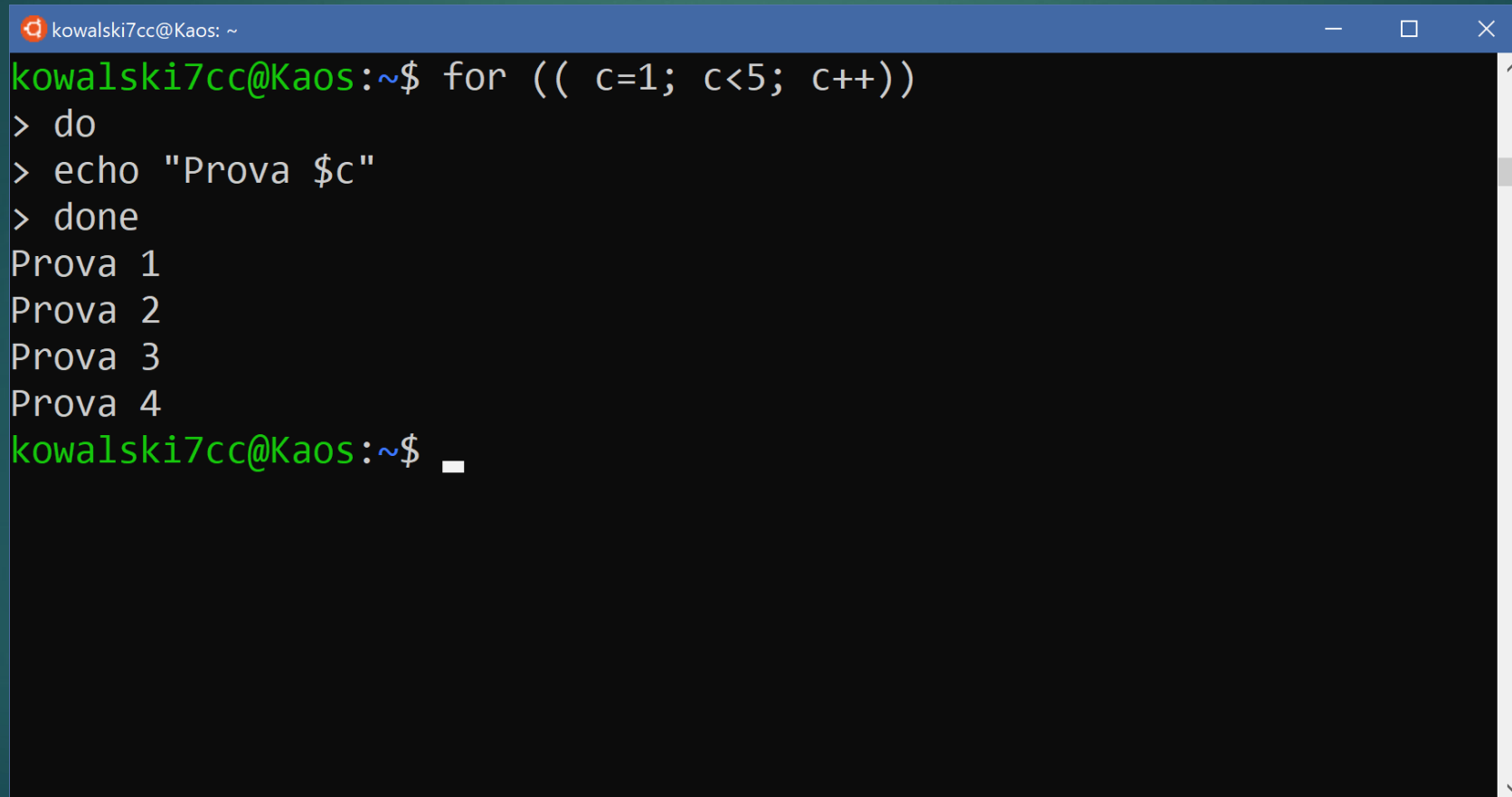
Ciclo for

```
$ for f in <seq>; do  
$   <commando>  
$   ...  
$ done
```

Ciclo for

```
kowalski7cc@Kaos: ~  
kowalski7cc@Kaos:~$ for i in file1 file2 file3; do echo $i; done  
file1  
file2  
file3  
kowalski7cc@Kaos:~$ for i in {1..5}; do echo $i; done  
1  
2  
3  
4  
5  
kowalski7cc@Kaos:~$ for i in {1..5..2}; do echo $i; done  
1  
3  
5  
kowalski7cc@Kaos:~$
```

Ciclo for



```
kowalski7cc@Kaos: ~  
kowalski7cc@Kaos:~$ for (( c=1; c<5; c++))  
> do  
> echo "Prova $c"  
> done  
Prova 1  
Prova 2  
Prova 3  
Prova 4  
kowalski7cc@Kaos:~$
```

A terminal window with a blue title bar containing the text "kowalski7cc@Kaos: ~" and standard window controls. The terminal has a black background with green text for the prompt and white text for the commands and output. The code shown is a bash for loop that iterates from c=1 to c=4, printing "Prova 1" through "Prova 4". The prompt returns to the shell after the loop completes.

Cicli con find e xargs

```
$ find . -iname \*.txt -exec chmod go-wx {} \;
```

```
$ find /tmp -iname \*.sh | xargs grep sudo
```


Ottenere il risultato di un comando

```
$ var=` <comando 1> `
```

```
$ var=$( <comando 2> )
```

Verificare l'esecuzione di un comando

```
$ <comando> && echo OK || echo Failed
```

```
$ <comando>
```

```
$ if [ $? -eq 0 ]; then <successo>; else <fail>; fi
```

```
$ if [[ output=$(some_command) ]]; then
```

```
> echo "Il risultato è: $output"
```

```
> fi
```

Input utente

```
$ echo -n "Inserisci username: "; read nome
```

```
Inserisci username: kowalski7cc
```

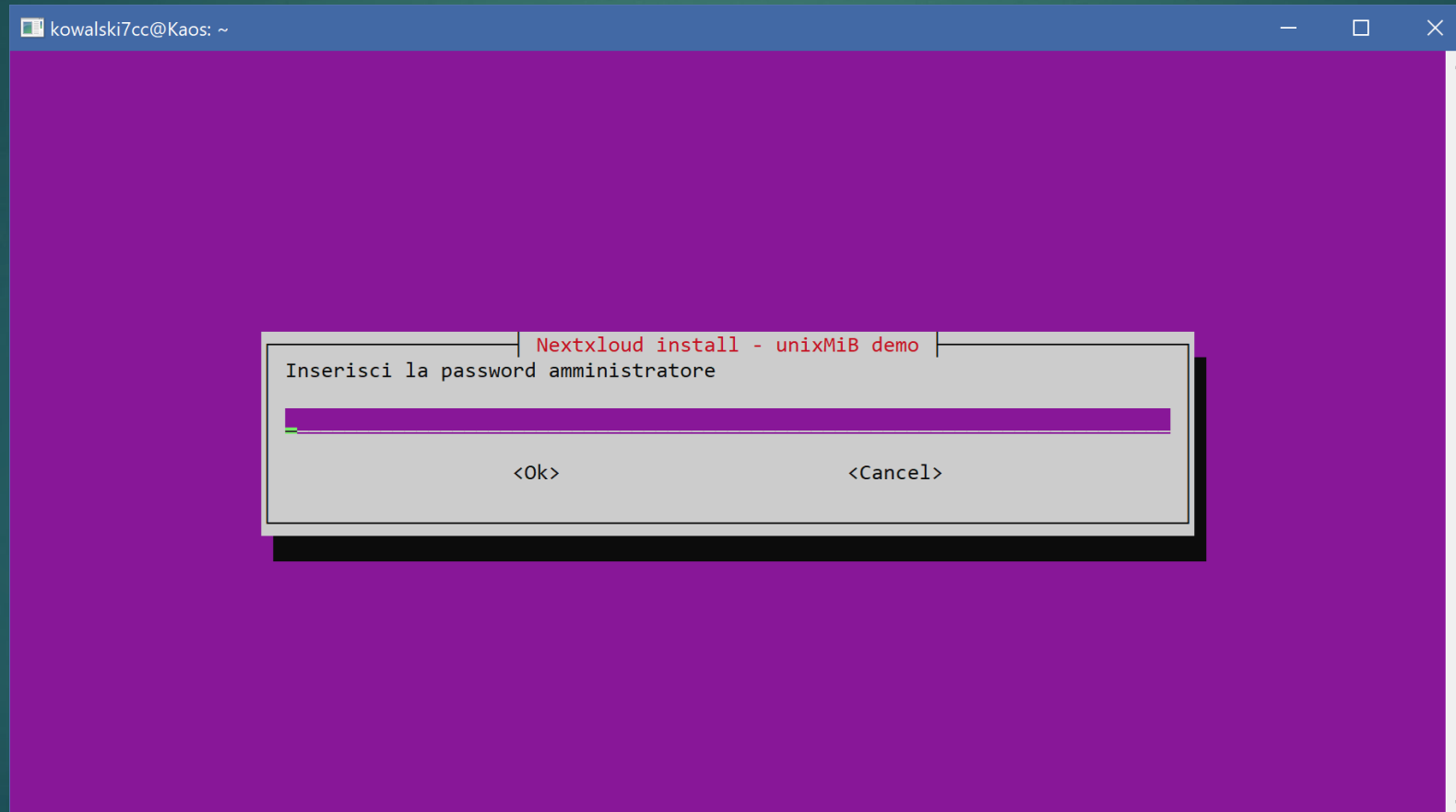
```
$ echo $nome
```

```
kowalski7cc
```

Input con "grafica"

```
$ whiptail --title "Example Dialog" --msgbox "This is an example of a  
message box. You must hit OK to continue." 8 78  
$ whiptail --title "Example Dialog" --yesno "This is an example of a  
yes/no box." 8 78  
$ whiptail --inputbox "What is your favorite Color?" 8 78 Blue --title  
"Example Dialog" 3>&1 1>&2 2>&3  
$ whiptail --passwordbox "please enter your secret password" 8 78 --  
title "password dialog" 3>&1 1>&2 2>&3
```

Input con "grafica"



Argomenti di uno script

```
$ ./script.sh parametro1 parametro2 "parametro3 con spazi"
```

Argomenti di uno script

\$0 Percorso enome dello script

\$1 Primo argomento

\$2 Secondo argomento

...

\$@ Tutti gli argomenti

\$# Il numero degli argomenti passati

Lavorare con i file

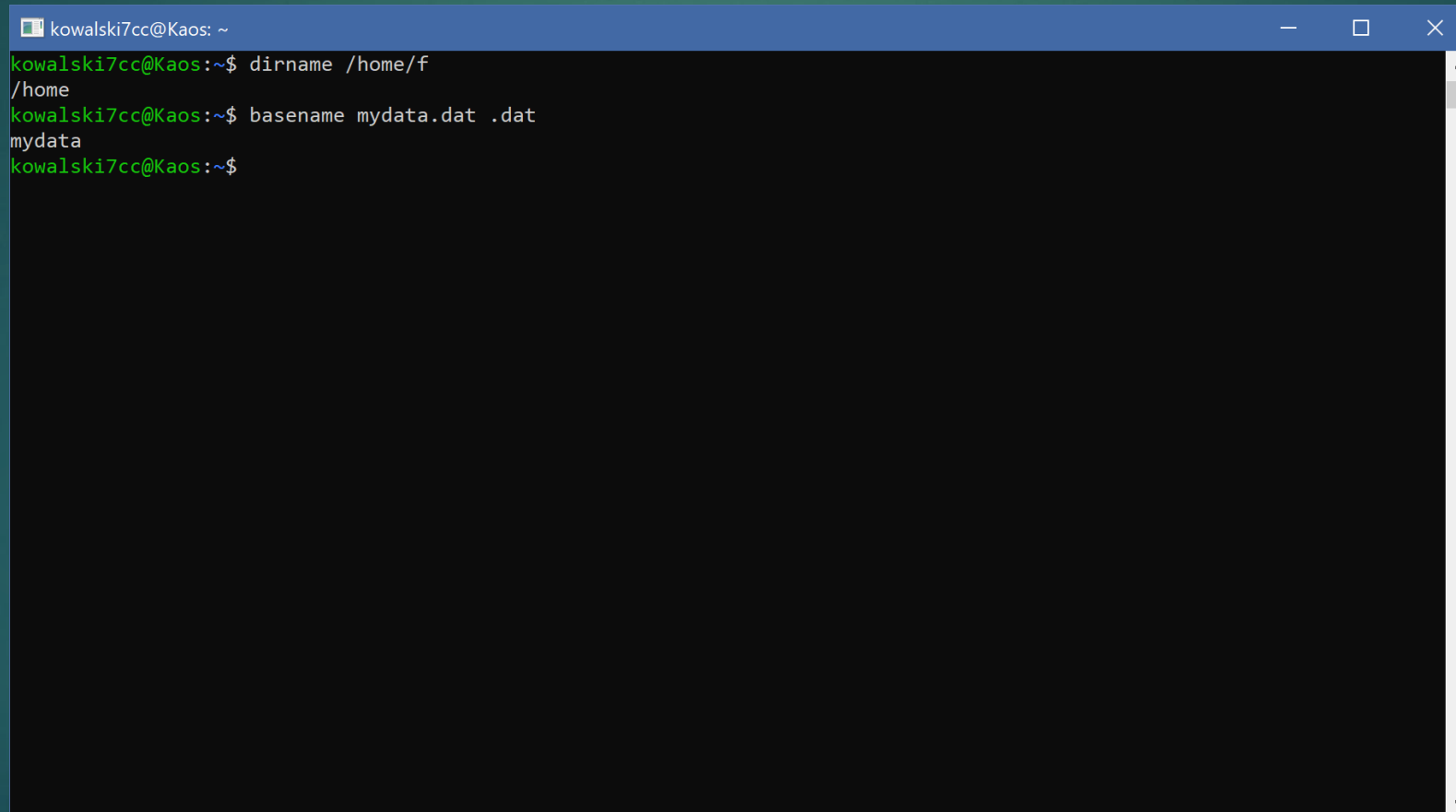
```
$ dirname path/file.f
```

```
path
```

```
$ basename file.ext .ext
```

```
file
```


Lavorare con i file

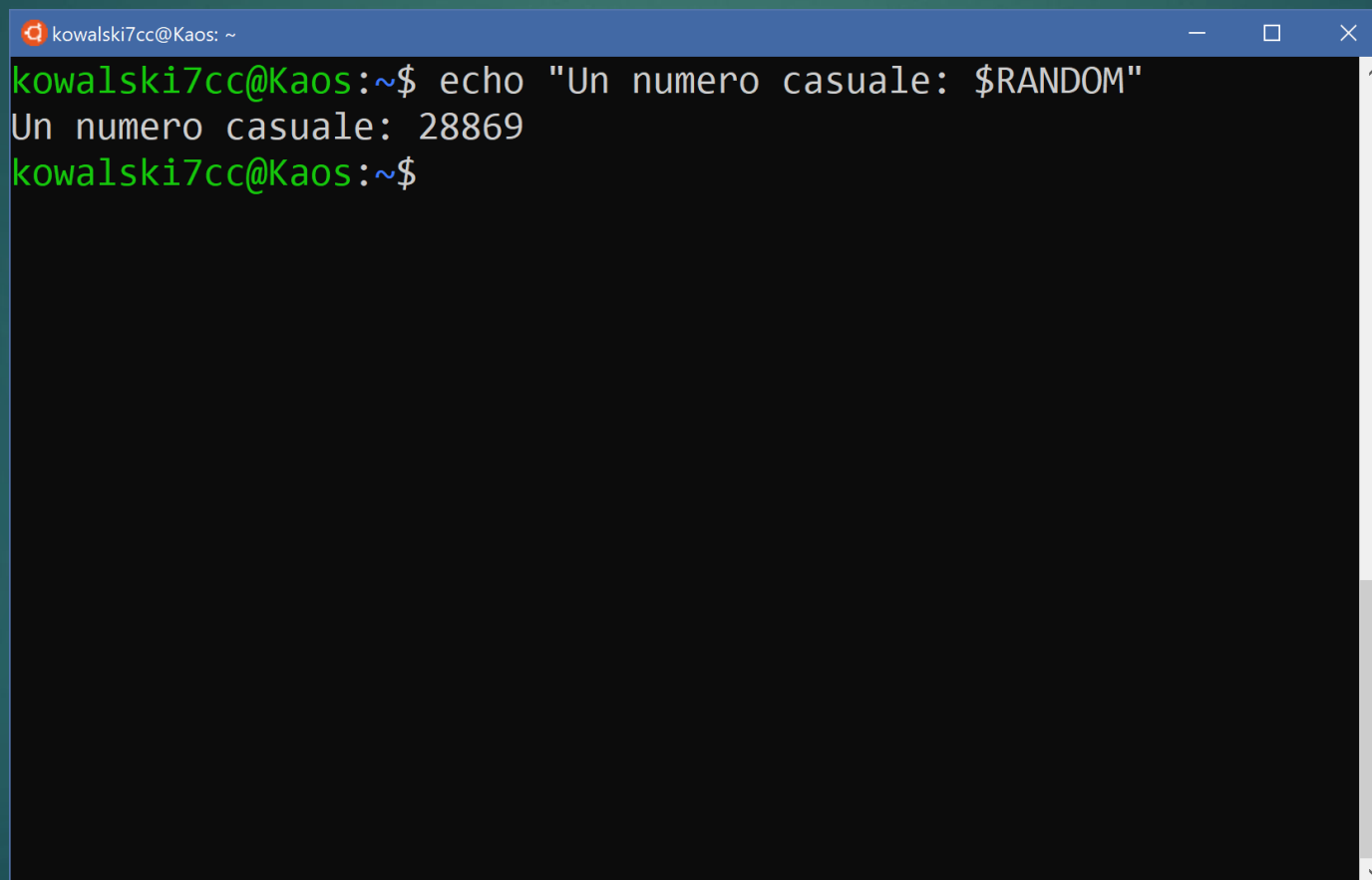
A terminal window with a blue title bar containing the text 'kowalski7cc@Kaos: ~' and standard window control icons. The terminal has a black background with green text. It shows the execution of two commands: 'dirname /home/f' which outputs '/home', and 'basename mydata.dat .dat' which outputs 'mydata'.

```
kowalski7cc@Kaos: ~  
kowalski7cc@Kaos:~$ dirname /home/f  
/home  
kowalski7cc@Kaos:~$ basename mydata.dat .dat  
mydata  
kowalski7cc@Kaos:~$
```

Numeri casuali in bash

```
$ # I numeri casuali in bash si trovano nella variabile random  
$ # $RANDOM contiene numeri interi a 16 bit [0-32767]  
$ echo $RANDOM  
<numero intero>
```

Numeri casuali in bash



```
kowalski7cc@Kaos: ~  
kowalski7cc@Kaos:~$ echo "Un numero casuale: $RANDOM"  
Un numero casuale: 28869  
kowalski7cc@Kaos:~$
```

A terminal window titled "kowalski7cc@Kaos: ~" with standard window controls. The prompt is "kowalski7cc@Kaos:~\$". The command "echo "Un numero casuale: \$RANDOM"" is entered and executed, resulting in the output "Un numero casuale: 28869". The prompt "kowalski7cc@Kaos:~\$" is shown again on the next line.

Socket in bash

```
$ # Apertura socket in bash
$ exec 3<>/dev/tcp/localhost/80
$ # Scrittura su socket
$ echo -e "data\r\n\r\n" >&3
$ # Lettura da socket
$ cat <&3
```

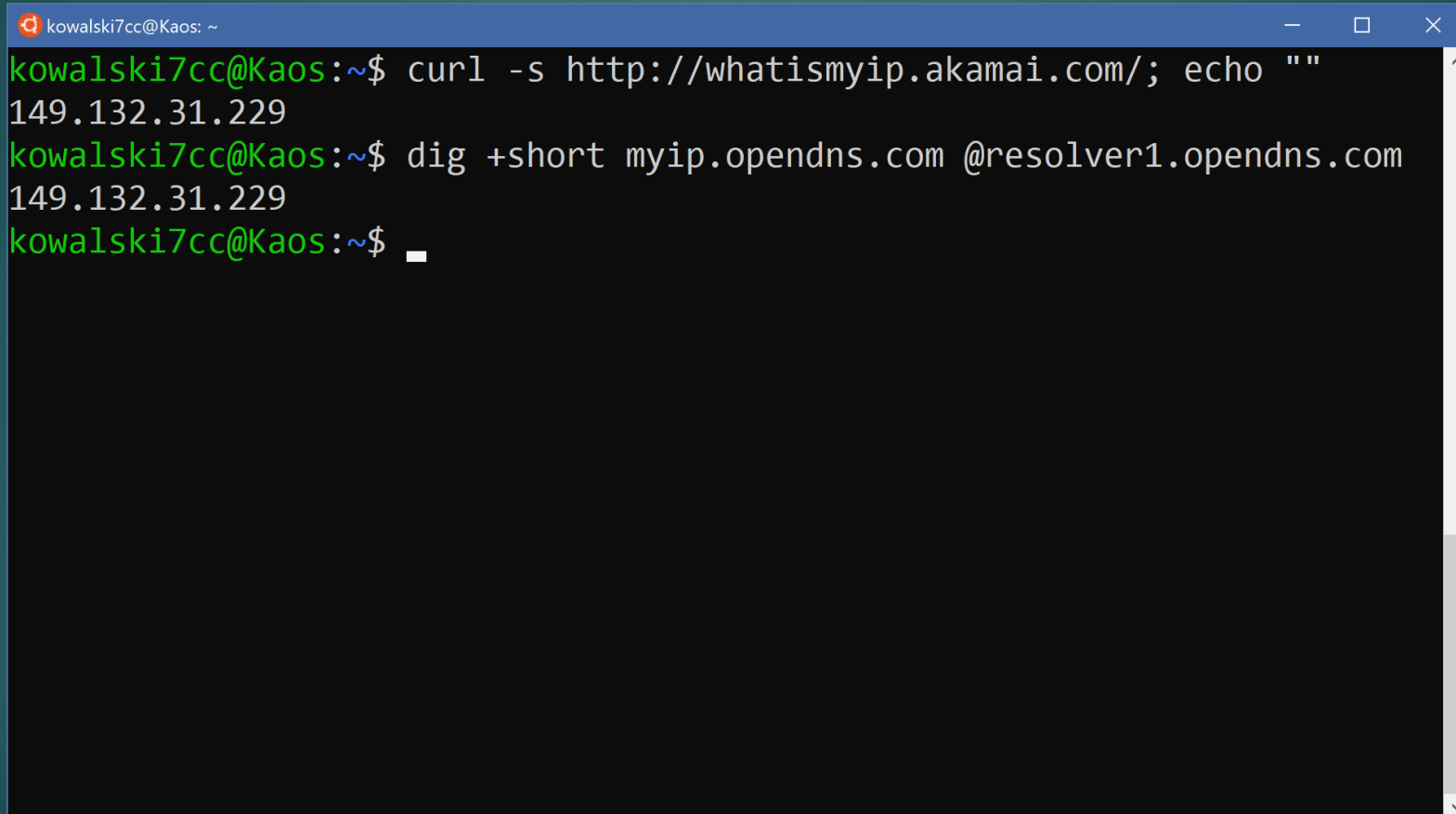
Ottenere dati da internet

```
$ curl <indirizzo>  
$ wget <indirizzo>  
$ telnet <indirizzo>
```

Ottenere dati da internet

```
kowalski7cc@Kaos: ~  
kowalski7cc@Kaos:~$ curl google.com  
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">  
<TITLE>301 Moved</TITLE></HEAD><BODY>  
<H1>301 Moved</H1>  
The document has moved  
<A HREF="http://www.google.com/">here</A>.  
</BODY></HTML>  
kowalski7cc@Kaos:~$
```

Ottenere dati da internet

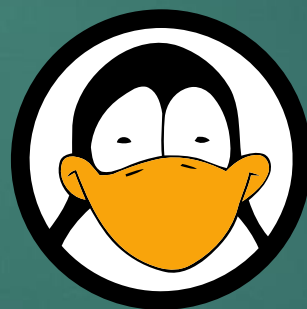
A terminal window with a blue title bar containing the text 'kowalski7cc@Kaos: ~' and standard window controls. The terminal has a black background with green text for the prompt and white text for the commands and output. The commands executed are 'curl -s http://whatismyip.akamai.com/; echo ""' and 'dig +short myip.opendns.com @resolver1.opendns.com', both of which return the IP address '149.132.31.229'. The window has a vertical scrollbar on the right side.

```
kowalski7cc@Kaos: ~  
kowalski7cc@Kaos:~$ curl -s http://whatismyip.akamai.com/; echo ""  
149.132.31.229  
kowalski7cc@Kaos:~$ dig +short myip.opendns.com @resolver1.opendns.com  
149.132.31.229  
kowalski7cc@Kaos:~$
```


Un grazie speciale a



STUDENTI ★
INDIPENDENTI



BGLUG
Bergamo Linux Users Group

Prossimi eventi

Cybersecurity: analisi forense
Venerdì 18 Maggio



Accessibilità
Mercoledì 23 Maggio



Seguici

- [facebook/unixMiB](https://facebook.com/unixMiB)
- [@unixMiB](https://twitter.com/unixMiB)
- unixmib.github.io

