

Corso di Sistemi Operativi
A.A. 2020/2021
Arzigogolo 4 – 16 ottobre 2020
Francesco Malferrari - Matricola 142795

Esercizio 1. Si scarichi l'immagine **landscape.jpg** e si risponda alle seguenti domande:

- quando è stata scattata la foto?
- dove è stata scattata la foto?
- con quale dispositivo è stata scattata la foto?
- chi ha scattato la foto?

Si assumano valide le eventuali informazioni trovate nell'immagine. Si documenti la procedura seguita.

Soluzione. Una volta scaricata l'immagine da <https://weblab.ing.unimore.it/people/andreolini/didattica/sistemi-operativi/index.html> si possono vedere la maggior parte delle informazioni richieste cliccando tasto destro sull'immagine, "Proprietà" e guardando nei dettagli:

- La foto è stata scattata il 06/11/2017 alle 12:48 (Figura 1);
- Le coordinate della latitudine e della longitudine (Figura 2) digitate su Google Maps (<https://www.google.com/maps/place/51%C2%B011'13.6%22N+1%C2%B047'07.3%22W/@51.1871143,-1.7875622,17z/data=!3m1!4b1!4m5!3m4!1s0x0:0x0!8m2!3d51.187111!4d-1.7853735>) portano a dichiarare che è stata scattata ad Amesbury, Salisbury, Regno Unito;
- Il dispositivo usato è una fotocamera Panasonic DMC-TZ60 (Figura 3);
- L'autore della foto non si trova scritto direttamente, è codificato in Base64 e si trova nell'ultima riga delle stringhe contenute nell'immagine, visibili attraverso il comando (Figura 4):

strings landscape.jpg

Per decodificare l'informazione bisogna mettere in pipe due comandi: uno per passare la stringa da analizzare e una per decodificarla.

In questo caso la stringa passata è l'ultima riga stampata dal comando precedente.

echo "TWF1cm8gQW5kcmVvbGluaQ==" | base64 -d

Alla fine risulta che "Mauro Andreolini" ha scattato questa foto (Figura 5).

```
Origine
-----
Autori
Data acquisizione      06/11/2017 12:48
Nome programma         paint.net 4.0.19
Data trasferimento
Copyright
```

Figura 1: Sezione "Origine" di **landscape.jpg**

```
GPS
-----
Latitudine      51; 11; 13.59930555554...
Longitudine     1; 47; 7.3446905167521...
```

Figura 2: Sezione "GPS" di **landscape.jpg**

Fotocamera

Produttore fotocamera	Panasonic
Modello fotocamera	DMC-TZ60
F-stop	f/3.3
Tempo di esposizione	1/1300 sec.
Sensibilità ISO	ISO-160
Esposizione	0 passaggio
Distanza focale	4 mm
Massima apertura	3.4453125
Modalità regolazione	Motivo
Distanza del soggetto	
Modalità flash	Nessun flash, obbligatorio
Energia flash	
lunghezza focale 35 mm	27

Figura 3: Sezione “Fotocamera” di *landscape.jpg*

```
studente@debian: ~
File Modifica Visualizza Cerca Terminale Aiuto
2Y-s
\[ $2%
hJjV
9#-nV
)/K-
YMcu8
aRqK
*G<e$
|Y}c
uwuh
{+20|
J2Mim
_l[y>
l%dx
_Iyt
=U-|Gh
NN:y
)gYR
9sYke
Twvw
ppeGr
.YG0
TWf1cm8gQW5kcmVvbGluaQ==
studente@debian:~$
```

Figura 4: Esecuzione del comando *strings landscape.jpg*

```
studente@debian: ~
File Modifica Visualizza Cerca Terminale Aiuto
studente@debian:~$ echo "TWf1cm8gQW5kcmVvbGluaQ==" | base64 -d
Mauro Andreolini
studente@debian:~$
```

Figura 5: Decodifica dell'ultima riga di *strings landscape.jpg* che risulta l'autore “Mauro Andreolini”

Esercizio 2. Si generi un file di dimensione pari a 50MB e contenuto a libera scelta. Si trovino due modi distinti per copiare il file in un altro con le seguenti modalità:

- limitazione della velocità di copia a 500KB/s;
- stampa di un indicatore di progresso.

Si documentino le procedure individuate e si evidenzino eventuali difficoltà di natura tecnica.

Soluzione. Per creare un file appositamente di dimensione 50MB si può utilizzare il comando:

```
fallocate -l 50M file.txt
```

dove 50M sta per 50MB.

Mentre, per procedere alla risoluzione dell'esercizio, la prima strategia è usare il comando da digitare sulla bash scritto qua sotto:

```
pv -p -L 512000 file.txt > ./condivisione
```

Il parametro **-L** consente di fissare il limite di velocità di copia, che adesso è fissato a 512000 byte al secondo (cioè 500KB/s).

-p, invece, stampa l'indicatore di progresso.

Il comando in esecuzione è visibile nella Figura 6.

La seconda strategia consiste invece nell'usare **rsync** in questo modo:

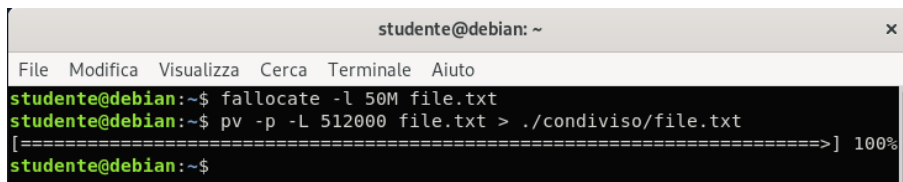
```
rsync -aPWh --bwlimit=500 file.txt ./condiviso
```

-aPWh permette di effettuare la copia con l'indicatore di progresso e **--bwlimit** fissa il limite di velocità dell'operazione in KB/s e quindi va passato il valore 500.

Il comando è visibile in esecuzione nella Figura 7.

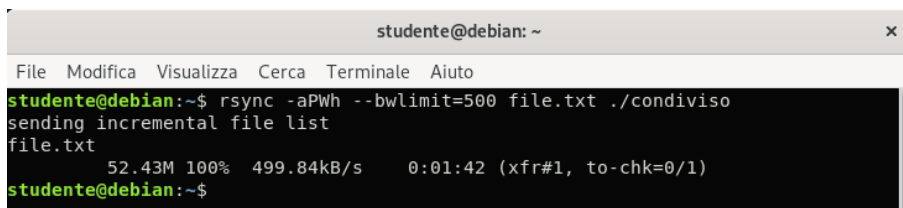
rsync e **pv** sono da installare.

Questi due metodi permettono di copiare il file **file.txt** nella cartella **condiviso**.



```
studente@debian: ~  
File Modifica Visualizza Cerca Terminale Aiuto  
studente@debian:~$ fallocate -l 50M file.txt  
studente@debian:~$ pv -p -L 512000 file.txt > ./condiviso/file.txt  
[=====] 100%  
studente@debian:~$
```

Figura 6: Soluzione Esercizio 2 e creazione file di 50MB



```
studente@debian: ~  
File Modifica Visualizza Cerca Terminale Aiuto  
studente@debian:~$ rsync -aPWh --bwlimit=500 file.txt ./condiviso  
sending incremental file list  
file.txt  
52.43M 100% 499.84kB/s 0:01:42 (xfr#1, to-chk=0/1)  
studente@debian:~$
```

Figura 7: Soluzione Esercizio 2

Esercizio 3. Si scriva uno script di shell **ssh-agent** che svolge le seguenti operazioni:

- dorme per un minuto;
- prende uno screenshot tramite Webcam;
- invia l'immagine ad un computer remoto (può essere lo stesso usato per scrivere lo script).

Si faccia in modo che il programma parta ogni volta che l'utente apre un emulatore di terminale.

Si usino esclusivamente strumenti da linea di comando (è concessa l'installazione di un convertitore video). Si commenti la procedura eseguita. Sotto quali condizioni funziona?

Soluzione. La risoluzione dell'esercizio è stata svolta attraverso il file shell **ssh-agent** che contiene i comandi:

- **sleep 60** che mette in pausa la bash per 60 secondi;
- **ffmpeg** che è un convertitore video che permette anche di realizzare screenshot tramite webcam attraverso l'opzione **-vframes 1** (cioè registra solo un frame). In questo caso la risoluzione è stata fissata a 640x480, ma è modificabile anche in base alle potenzialità del device di registrazione;
- **nc** che serve per stabilire un meccanismo di trasferimento non autenticato con un altro dispositivo. Nel file il comando serve per inviare **immagine.jpg** all'indirizzo localhost alla porta 1234 in questo caso. Tuttavia per stabilire il collegamento serve la presenza di un server che attenda il file inviato nella specifica porta e lo acquisisca, questo si stabilisce attraverso questo comando (Figura 8):

```
nc -l -p 1234 > ./condiviso/immagine.jpg
```

Il file verrà posizionato nella cartella **./condiviso**.

Il contenuto di **ssh-agent** è il seguente (è visibile anche nella cartella dove è presente questo documento):

```
sleep 60
ffmpeg -f video4linux2 -s 640x480 -i /dev/video0 -vframes 1
immagine.jpg
nc -w 3 127.0.0.1 1234 < immagine.jpg
```

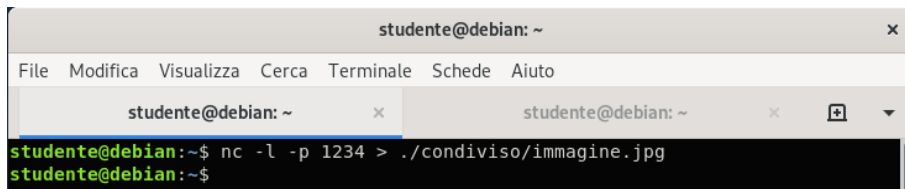
Per poter fare in modo che tutto questo vada in esecuzione ogni volta che un emulatore di terminale viene aperto (Figura 9) è stato aggiunto alla fine del file **~/.bashrc** il comando per eseguire **ssh-agent**. (Figura 10).

Il tutto funziona se si dispone di una webcam collegata al dispositivo, letta dalla macchina Debian che posiziona le sue configurazioni all'interno di **/dev/video0**.

Per stabilire la connessione tra client e server bisogna che sia in esecuzione prima il server in una finestra a parte e poi **ssh-agent**, in caso contrario il tentativo di trasferimento fallirà.

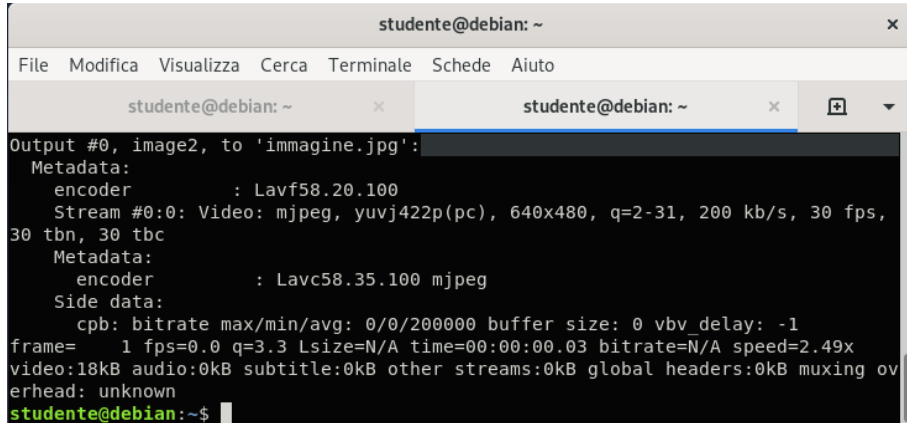
In questo caso, dove il dispositivo di destinazione è lo stesso che manda il file, non è necessaria la connessione a Internet, ma in altre situazioni è indispensabile per stabilire la comunicazione.

Solo con queste condizioni è possibile portare a termine l'esecuzione corretta di **ssh-agent**.



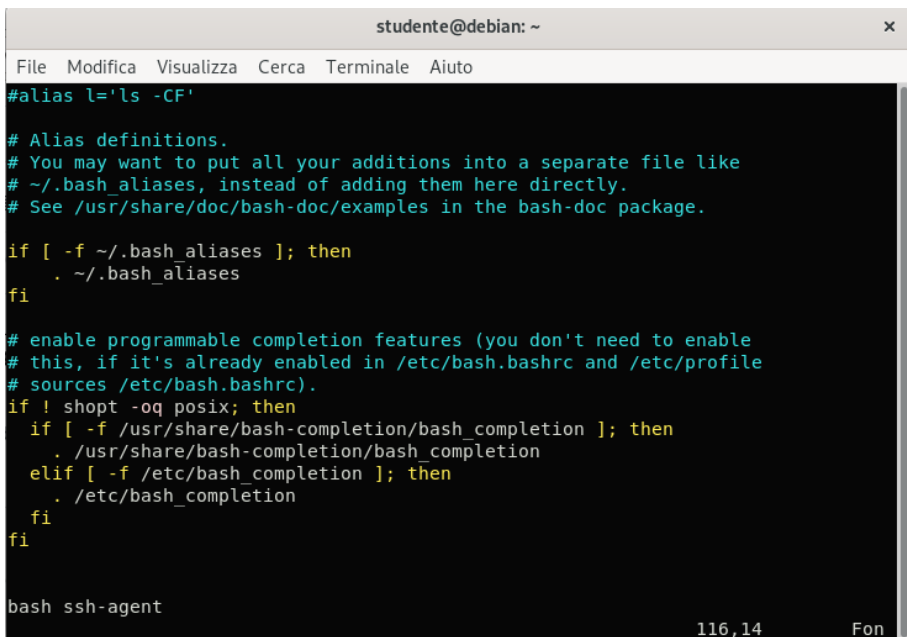
```
studente@debian: ~  
File Modifica Visualizza Cerca Terminale Schede Aiuto  
studente@debian: ~ x studente@debian: ~ x  
studente@debian:~$ nc -l -p 1234 > ./condiviso/immagine.jpg  
studente@debian:~$
```

Figura 8: Finestra del server



```
studente@debian: ~  
File Modifica Visualizza Cerca Terminale Schede Aiuto  
studente@debian: ~ x studente@debian: ~ x  
Output #0, image2, to 'immagine.jpg':  
  Metadata:  
    encoder      : Lavf58.20.100  
    Stream #0:0: Video: mjpeg, yuvj422p(pc), 640x480, q=2-31, 200 kb/s, 30 fps,  
30 tbn, 30 tbc  
  Metadata:  
    encoder      : Lavc58.35.100 mjpeg  
  Side data:  
    cpb: bitrate max/min/avg: 0/0/200000 buffer size: 0 vbv_delay: -1  
frame=   1 fps=0.0 q=3.3 Lsize=N/A time=00:00:00.03 bitrate=N/A speed=2.49x  
video:18kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing ov  
erhead: unknown  
studente@debian:~$
```

Figura 9: Apertura emulatore di terminale



```
studente@debian: ~  
File Modifica Visualizza Cerca Terminale Aiuto  
#alias l='ls -CF'  
  
# Alias definitions.  
# You may want to put all your additions into a separate file like  
# ~/.bash_aliases, instead of adding them here directly.  
# See /usr/share/doc/bash-doc/examples in the bash-doc package.  
  
if [ -f ~/.bash_aliases ]; then  
  . ~/.bash_aliases  
fi  
  
# enable programmable completion features (you don't need to enable  
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile  
# sources /etc/bash.bashrc).  
if ! shopt -oq posix; then  
  if [ -f /usr/share/bash-completion/bash_completion ]; then  
    . /usr/share/bash-completion/bash_completion  
  elif [ -f /etc/bash_completion ]; then  
    . /etc/bash_completion  
  fi  
fi  
  
bash ssh-agent  
116,14 Fon
```

Figura 10: Fine del file ~/.bashrc