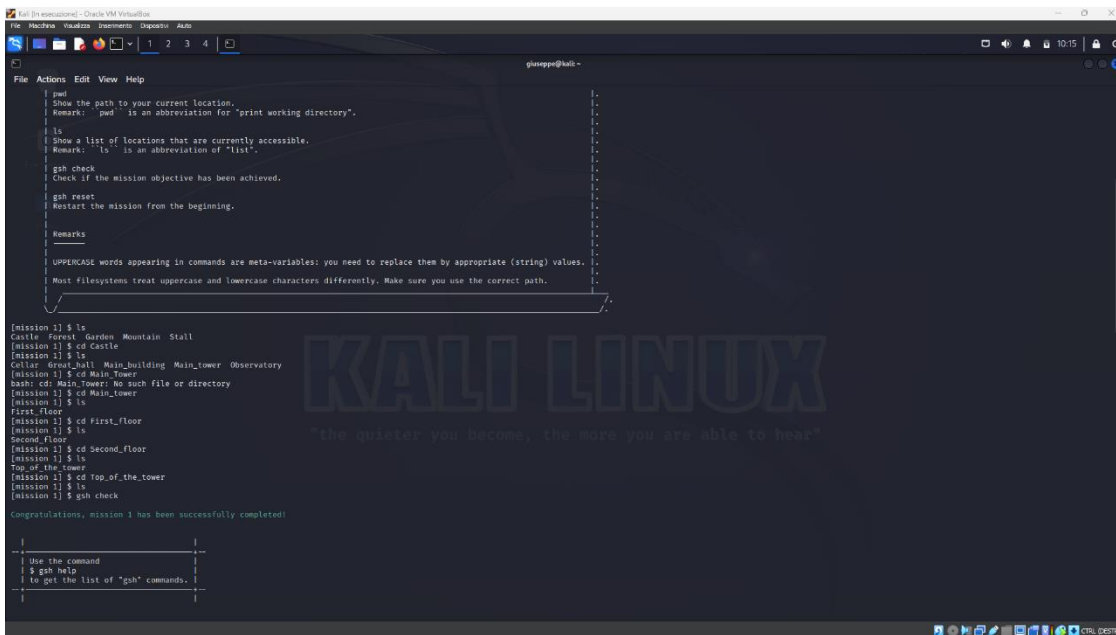


GameShell: analisi dei primi 10 livelli.

Di seguito, analizzeremo i primi 10 livelli di GameShell, spiegando nel dettaglio i comandi utilizzati.

Innanzitutto, GameShell è una distribuzione che permette agli utenti di “allenarsi” con i comandi dello shell di Linux utilizzando la forma di un gioco interattivo, dove ogni livello per essere completato necessita di una determinata sequenza di codici. Per brevità e per mostrarne il funzionamento, esamineremo nel dettaglio i primi 10 livelli; per ognuno di questi, utilizziamo il comando **gsh goal** per conoscere l’obiettivo e, una volta raggiunto, il comando **gsh check** per completarli.



```
Kali Linux
File Actions Edit View Help

| pwd |
| Show the path to your current location. |
| Remark: 'pwd' is an abbreviation for 'print working directory'. |
| ls |
| Show a list of locations that are currently accessible. |
| Remark: 'ls' is an abbreviation of 'list'. |
| gsh check |
| Check if the mission objective has been achieved. |
| gsh reset |
| Restart the mission from the beginning. |
| Remarks |
| UPPERCASE words appearing in commands are meta-variables; you need to replace them by appropriate (string) values. |
| Most filesystems treat uppercase and lowercase characters differently. Make sure you use the correct path. |

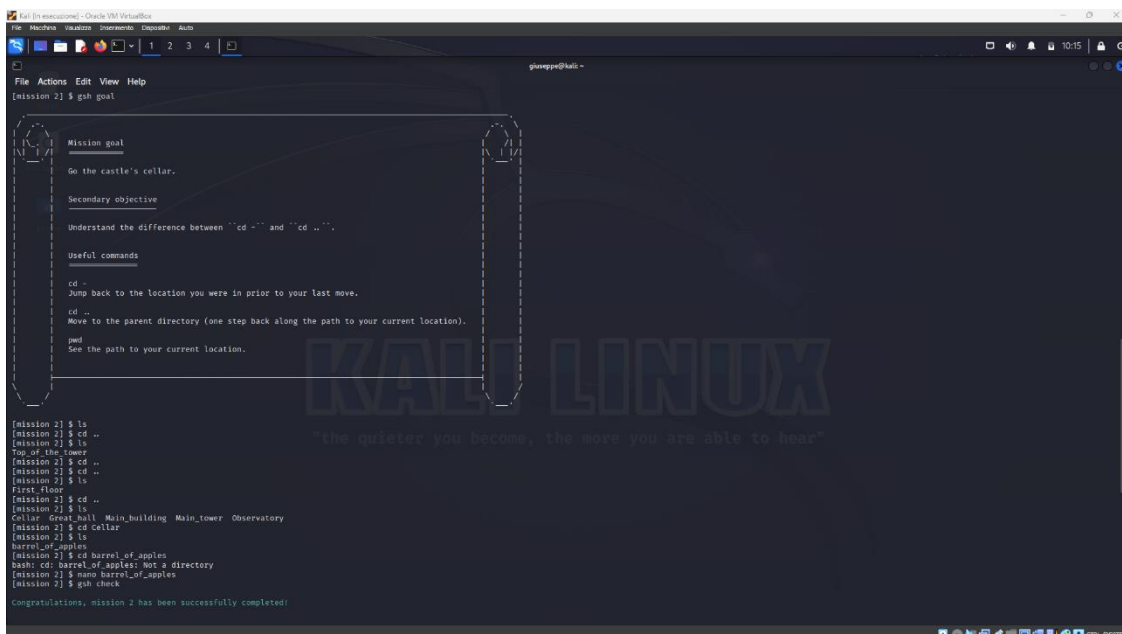
[mission 1] $ ls
Castle Forest Garden Mountain Stall
[mission 1] $ cd Castle
[mission 1] $ ls
Cellar Great_Hall Main_building Main_tower Observatory
[mission 1] $ cd Main_Tower
bash: cd: Main_Tower: no such file or directory
[mission 1] $ cd Main_tower
[mission 1] $ ls
First_floor
[mission 1] $ cd First_floor
[mission 1] $ ls
Second_floor
[mission 1] $ cd Second_floor
[mission 1] $ ls
Top_of_the_tower
[mission 1] $ cd Top_of_the_tower
[mission 1] $ ls
[mission 1] $ gsh check

Congratulations, mission 1 has been successfully completed!

| Use the command |
| $ gsh help |
| To get the list of "gsh" commands. |
```

Nel primo livello ci viene chiesto di raggiungere la cima della torre del castello, ossia una directory chiamata **Top_of_the_tower**. Facciamo ciò esaminando il contenuto di ogni directory in cui ci troviamo con il comando **ls**, spostandoci poi in quella necessaria con **cd**. Alternando i comandi come mostrato in figura,

raggiungiamo l’obiettivo e confermiamo con **gsh check**.



```
Kali Linux
File Actions Edit View Help

[mission 2] $ gsh goal

Mission goal
Go the castle's cellar.

Secondary objective
Understand the difference between "cd -" and "cd ..".

Useful commands
cd -
Jump back to the location you were in prior to your last move.
cd ..
Move to the parent directory (one step back along the path to your current location).
pwd
See the path to your current location.

[mission 2] $ ls
[mission 2] $ cd -
[mission 2] $ ls
Top_of_the_tower
[mission 2] $ cd ..
[mission 2] $ cd ..
[mission 2] $ ls
First_floor
[mission 2] $ cd ..
[mission 2] $ ls
Cellar Great_Hall Main_building Main_tower Observatory
[mission 2] $ cd cellar
[mission 2] $ ls
barrel_of_apples
[mission 2] $ cd barrel_of_apples
bash: cd: barrel_of_apples: not a directory
[mission 2] $ pwd
barrel_of_apples
[mission 2] $ gsh check

Congratulations, mission 2 has been successfully completed!
```

Nel secondo livello, implementiamo i comandi **cd -** e **cd ..** per spostarci nelle directory precedentemente visitate o nella directory padre rispetto alla nostra attuale posizione. Abbiamo inoltre il comando **pwd** per verificare la nostra posizione relativa alle directory.

The screenshot shows a Kali Linux terminal window with the following content:

```
File Actions Edit View Help
$ cd
to go back to the GameShell's starting directory.
[mission 2] $ cd
[mission 2] $ gsh reset

|-----|
| Use the command |
| $ gsh help      |
| to get the list of "gsh" commands. |
|-----|

[mission 3] $ gsh goal

Mission goal
Go back to the starting location and then go to the throne room using only two commands.
Remark
You may experiment with as many commands as you want, but
to validate the mission the following conditions need to be met:
- the second to last command takes you to the starting point,
- the last command takes you directly to the throne room.
Useful commands
cd
Move back to the starting location.
cd (LOCATION1/LOCATION2/LOCATION3)
Make several moves in one command.
Remark
UPPERCASE words appearing in commands are meta-variables: you need to replace them by appropriate (string) values.

[mission 2] $ cd
[mission 2] $ cd /home/giuseppe/gameshell/World/Castle/Main_building/throne_room
[mission 2] $ gsh check
Congratulations, mission 3 has been successfully completed!
```

Nel livello 3, specifichiamo direttamente in linea di comando la directory nella quale desideriamo spostarci, attraverso il comando `cd` *~path/directory_interessata*, dove path rappresenta il percorso e *directory_interessata* la cartella in cui desideriamo spostarci, come de esempio in figura.

The screenshot shows a Kali Linux terminal window with the following content:

```
File Actions Edit View Help
|-----|
| Use the command |
| $ gsh help      |
| to get the list of "gsh" commands. |
|-----|

~/Castle/Main_building/Throne_room
[mission 4] $ gsh goal

Mission goal
Build a "Hut" in the forest, and then build a "Chest" in the hut.
Useful commands
mkdir DIRECTORY
Create a new directory inside the current directory.
Remark: "mkdir" is an abbreviation for "Make directory".

~/Castle/Main_building/Throne_room
[mission 4] $ cd /home/giuseppe/gameshell/World/Forest
~/Forest
[mission 4] $ mkdir Hut
~/Forest
[mission 4] $ cd Hut
~/Forest/Hut
[mission 4] $ mkdir Chest
~/Forest/Hut
[mission 4] $ gsh check
Congratulations, mission 4 has been successfully completed!
```

Nel livello 4, usando il comando **mkdir**, creiamo nuove directory nella posizione in cui ci troviamo, una dentro l'altra.

The screenshot shows a Kali Linux terminal window with the following content:

```
File Actions Edit View Help
[mission 4] $ gsh check
Congratulations, mission 4 has been successfully completed!

|-----|
| Use the command |
| $ gsh help      |
| to get the list of "gsh" commands. |
|-----|

~/Forest/Hut
[mission 5] $ gsh goal

Mission goal
Go back to the cellar and get rid of all the spiders. Leave the bats alone: they appear on the castle's coat of arms and are said to confer luck.
Useful commands
rm FILE1 FILE2 ... FILEn
Delete the files (permanently).
Remark: "rm" is an abbreviation for "remove".

~/Forest/Hut
[mission 5] $ /home/giuseppe/gameshell/World/Castle/cellar
bash: /home/giuseppe/gameshell/World/Castle/cellar: is a directory
~/Forest/Hut
[mission 5] $ cd /home/giuseppe/gameshell/World/Castle/cellar
~/Castle/cellar
[mission 5] $ ls
barrel_of_spices bat.1 bat.2 spider.1 spider.2 spider.3
~/Castle/cellar
[mission 5] $ rm spider.1 spider.2 spider.3
~/Castle/cellar
[mission 5] $ gsh check
Congratulations, mission 5 has been successfully completed!
```

Nel livello 5, ci viene chiesto di eliminare dei file, i “ragni”, nella directory Cellar: facciamo ciò con il comando **rm** seguito del nome dei file da eliminare. Vedremo successivamente come possiamo ottimizzare questa operazione con *****.

```
Kali Linux VM Window
File Machine Viewports Disposed Auto
1 2 3 4
g.iuseppe@kali ~
File Actions Edit View Help
Useful commands
mv FILES FILE2 ... FILE: DIRECTORY
Move the FILES to the directory.
Remark: "mv" is an abbreviation of "move".
The "-" symbol is an abbreviation for the initial directory.
Example: wherever you are, "--/Tavern" denotes the directory (or file) "Tavern" in the initial directory.
~/Castle/Cellar
[mission 6] $ /home/giuseppe/gameshell/World/Castle
bash: /home/giuseppe/gameshell/World/Castle: Is a directory
~/Castle/Cellar
[mission 6] $ ls
barrel_of_apples bat_1 bat_2
~/Castle/Cellar
[mission 6] $ cd /home/giuseppe/gameshell/World/Castle
~/Castle
[mission 6] $ ls
Cellar Great_hall Main_building Main_tower Observatory
~/Castle
[mission 6] $ cd ..
~/
[mission 6] $ ls
Castle Forest Garden Mountain Stall
~/
[mission 6] $ cd Garden
~/Garden
[mission 6] $ ls
Flower_garden Maze Shed coin_1 coin_2 coin_3
~/Garden
[mission 6] $ mv coin_1 coin_2 coin_3 /home/giuseppe/gameshell/World/Forest/Hut/Chest
~/Garden
[mission 6] $ gsh check
Congratulations, mission 6 has been successfully completed!
```

Nel livello 6, ci muoviamo nella directory Garden e il nostro obiettivo è spostare dei file, le “monete”. Utilizziamo il comando **mv** seguito dagli oggetti da spostare e infine dalla directory in cui vogliamo mettere i file, come mostrato in figura.

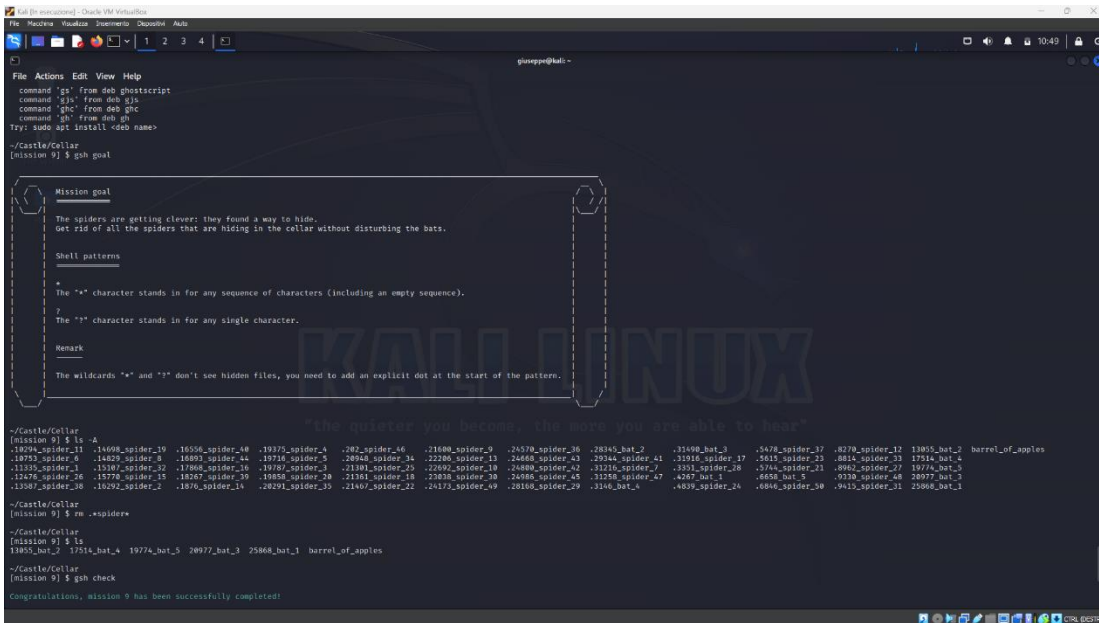
```
Kali Linux VM Window
File Machine Viewports Disposed Auto
1 2 3 4
g.iuseppe@kali ~
File Actions Edit View Help
Use the command
$ gsh help
to get the list of "gsh" commands.
~/Garden
[mission 7] $ gsh goal
()=(
Mission goal
Collect all the coins hidden in the garden in front of the castle, and put them in your chest (in your hut in the forest).
Secondary objective
Learn how to use the "Tab" key to go faster.
Useful commands
ls -A
List all the files of the current directory, including hidden files. (A file is "hidden" when its name starts with a dot.)
Tab
The tabulation key "completes" the name of a file or directory once you have typed the beginning of its name. This only works if there is only one possible completion.
Tab-Tab
Pressing tabulation twice successively shows a list of possible completions.
()=(
~/Garden
[mission 7] $ ls -A
. .16474_coin_2 .4735_coin_1 Flower_garden Maze Shed
~/Garden
[mission 7] $ mv .16474_coin_2 .4735_coin_1 .16148_coin_3 /home/giuseppe/gameshell/World/Forest/Hut/Chest
~/Garden
[mission 7] $ gsh check
Congratulations, mission 7 has been successfully completed!
```

Nel livello 7, scopriamo che alcune “monete” (ossia alcuni file) sono nascoste: usiamo **ls -A** per visualizzarle e vedere il loro nome e ancora una volta il comando **mv** per spostarle.

Aggiungiamo la funzionalità del tasto **Tab** per non dover scrivere per intero il nome del file, ma solo dal momento in cui ciò che abbiamo digitato identifica un file univoco.

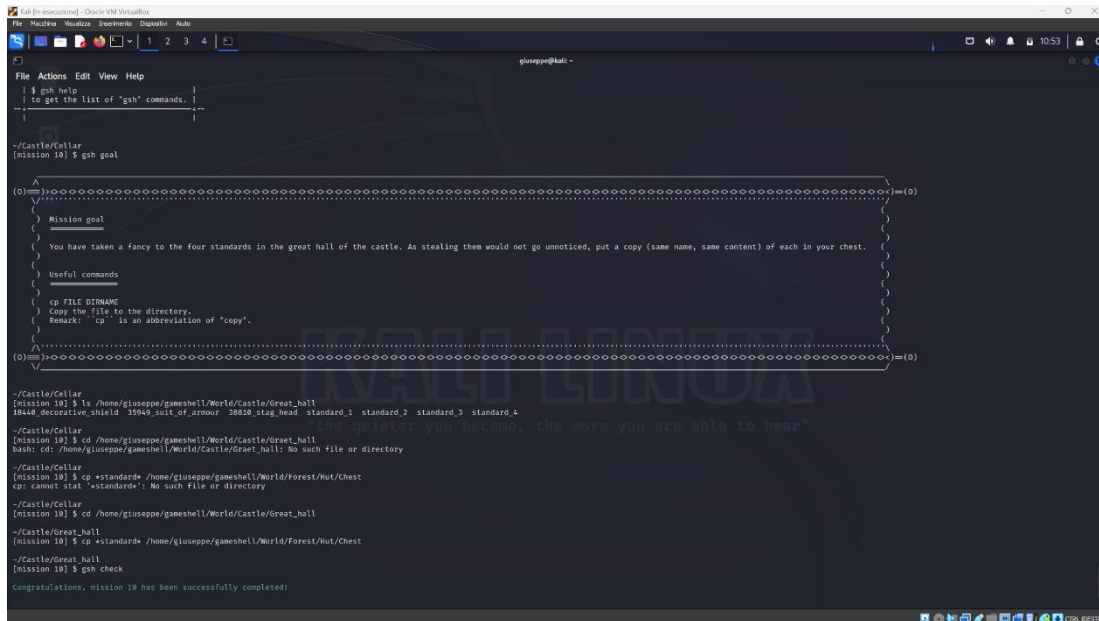
```
Kali Linux VM Window
File Machine Viewports Disposed Auto
1 2 3 4
g.iuseppe@kali ~
File Actions Edit View Help
Get rid of all the spiders that are crawling in the cellar. Again, do not do not disturb the bats.
Shell patterns
The "*" character stands in for any sequence of characters (including an empty sequence).
The "?" character stands in for any single character.
These wildcards can be used to denote lists of existing files / directories in the current working directory.
For example: if the current folder contains
file=1 Folder=1 file=14 potato
then
+1 → file=1 Folder=1 file=14 potato
+1 → file=1 Folder=1
+* → error: no matching file
+? → file=1 Folder=1
+* → file=14
~/Garden
[mission 8] $ cd /home/giuseppe/gameshell/World/Castle/Cellar
~/Castle/Cellar
[mission 8] $ ls
18461_spider_27 12665_spider_38 15227_spider_11 16679_spider_37 17631_spider_49 18348_spider_41 20604_spider_8 23275_spider_2 26378_spider_12 28395_spider_34 30938_spider_26 3584_spider_45 6528_spider_58 8899_spider_6
11925_spider_28 12895_bat_2 15372_spider_48 16921_spider_28 18887_spider_15 19879_spider_19 20977_bat_3 23648_spider_3 27846_spider_46 29945_spider_9 315_spider_43 3952_spider_16 8846_spider_14 8844_spider_1
119_spider_27 14996_spider_17 16389_spider_12 16958_spider_33 18194_spider_38 19346_spider_18 21832_spider_21 2581_spider_13 27774_spider_29 30388_spider_19 3241_spider_44 44878_spider_47 8814_spider_14 941_spider_4
12228_spider_18 15114_spider_31 16622_spider_7 17514_bat_4 18233_spider_15 19774_bat_5 2282_spider_23 25868_bat_1 28184_spider_16 30929_spider_45 33448_spider_39 4391_spider_5 8818_spider_43 barrel_of_apples
~/Castle/Cellar
[mission 8] $ rm *spider
rm: cannot remove '*spider': No such file or directory
~/Castle/Cellar
[mission 8] $ rm *spider
~/Castle/Cellar
[mission 8] $ ls
11855_bat_2 17514_bat_4 19774_bat_5 20977_bat_3 25868_bat_1 barrel_of_apples
~/Castle/Cellar
[mission 8] $ gsh check
Congratulations, mission 8 has been successfully completed!
```

Nel livello 8, ci viene chiesto di sbarazzarci di tutti i “ragni” presenti senza però interferire con i “pipistrelli”. Usiamo quindi **rm *spider***: in questo modo, andrò ad agire su tutti i file aventi *spider* nel nome, senza dover scrivere a mano il nome di ogni singolo elemento.



Nel livello 9, diamo ancora la caccia ad alcuni file “ragni” che si erano nascosti: li visualizziamo con **ls -A** e li eliminiamo con lo stesso comando usato in precedenza a cui però aggiungiamo un punto per permettergli di eliminare file nascosti, come da seguente sintassi:

rm *.spider*, rimuovendoli tutti in un colpo solo.



Infine, nel livello 10, ci viene chiesto di “rubare” degli standardi copiandoli nella directory Chest che avevamo creato in precedenza. Utilizziamo il comando **cp *standards*** seguito dalla directory dove ci interessa copiare gli elementi. Ancora una volta, utilizziamo ***testo*** per non dover inserire manualmente il nome di ogni singolo file che ci interessa, dato che tutti hanno la parte di testo specificata tra gli asterischi nel nome.

Il gioco continua ancora, fornendo ulteriori esempi e prove sull’utilizzo del codice; di seguito ad esempio uno screenshot del livello 31.

