

ESERCITAZIONE DI FINE

MODULO 2 - EPICODE

In questa esercitazione si è affrontato il gioco didattico GameShell: L'obiettivo è stato completare una serie di livelli che richiedevano l'utilizzo della shell Unix/Linux per navigare filesystem, manipolare file e cartelle, e imparare/perfezionare le conoscenze della shell.

```
+-----+
| Run the command
|   $ gsh goal
| to discover your first mission.
|
| You can check the mission has been
| completed with
|   $ gsh check
|
| The command
|   $ gsh help
| displays the list of available (gsh)
| commands.
+-----+
```

Livello 1: Utilizzo base del `cd`

Obiettivo della missione:

Mission goal

Go to the top of the main tower of the castle.

La prima missione richiede di raggiungere la cima della torre principale del castello. Per questo è stato necessario navigare l'albero delle directory con l'uso di `cd`.

Procedimento:

```
[use 'gsh help' to get a list of available commands]
[mission 1] $ ls
Castle  Forest  Garden  Mountain  Stall

[use 'gsh help' to get a list of available commands]
[mission 1] $ cd Castle

[use 'gsh help' to get a list of available commands]
[mission 1] $ ls
Cellar  Great_hall  Main_building  Main_tower  Observatory

[use 'gsh help' to get a list of available commands]
[mission 1] $ cd Main_tower

[use 'gsh help' to get a list of available commands]
[mission 1] $ ls
First_floor

[use 'gsh help' to get a list of available commands]
[mission 1] $ cd First_floor && ls
Second_floor

[use 'gsh help' to get a list of available commands]
[mission 1] $ cd Second_floor && ls
Top_of_the_tower

[use 'gsh help' to get a list of available commands]
[mission 1] $ cd Top_of_the_tower
```

```
[mission 1] $ gsh check
```

```
Congratulations, mission 1 has been successfully completed!
```

Analisi:

Il comando `cd` consente di navigare attraverso le directory, quindi serve a spostarsi liberamente nella shell. In questo caso, ho dovuto “esplorare” le directory: (Castle → Main_tower → First_floor → Second_floor → Top_of_the_tower) per raggiungere l'obiettivo. Con `gsh check` valutiamo lo stato attuale della missione.

In aggiunta ho usato semplici `ls` (list) per ispezionare l'ambiente.

Livello 2: Differenza tra `cd ..` e `cd -`

Obiettivo della missione:

```
Mission goal
=====

Go the castle's cellar.

Secondary objective
=====

Understand the difference between ``cd -`` and ``cd ..``.
```

La missione principale richiede di accedere alla cantina del castello. Inoltre, un obiettivo secondario (di apprendimento) chiede di comprendere la differenza tra `cd ..` e `cd -`.

Procedimento:

```
[use 'gsh help' to get a list of available commands]
[mission 2] $ cd -

[use 'gsh help' to get a list of available commands]
[mission 2] $ ls
Top_of_the_tower

[use 'gsh help' to get a list of available commands]
[mission 2] $ cd .. && ls
Second_floor

[use 'gsh help' to get a list of available commands]
[mission 2] $

[use 'gsh help' to get a list of available commands]
[mission 2] $ cd .. && ls
First_floor

[use 'gsh help' to get a list of available commands]
[mission 2] $ cd .. && ls
Cellar  Great_hall  Main_building  Main_tower  Observatory

[use 'gsh help' to get a list of available commands]
[mission 2] $ cd Cellar

[use 'gsh help' to get a list of available commands]
[mission 2] $ ls
barrel_of_apples

[use 'gsh help' to get a list of available commands]
[mission 2] $ gsh check

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

Analisi:

Durante questa missione ho testato la differenza tra due comandi spesso confusi:

- `cd ..`: sposta nella directory padre, ossia un livello più in alto nell'albero gerarchico del filesystem. È un movimento relativo alla posizione corrente.
- `cd -`: torna alla directory precedente, indipendentemente da dove si trova nell'albero. È essenzialmente un "undo" del terminale.

Questa distinzione è fondamentale per navigare efficientemente il filesystem, soprattutto quando si lavora in strutture complesse.

Struttura del filesystem

```
[mission 3] $ cd && tree
.
├── Castle
│   ├── Cellar
│   │   └── barrel_of_apples
│   ├── Great_hall
│   ├── Main_building
│   │   ├── Library
│   │   │   ├── Greek_Latin_and_other_modern_languages
│   │   │   ├── Mathematics_101
│   │   │   └── Merlin_s_office
│   │   │       ├── candle
│   │   │       └── Drawer
│   │   │           └── ink_and_scroll
│   │   ├── Throne_room
│   │   │   └── Kings_quarter
│   ├── Main_tower
│   │   ├── First_floor
│   │   │   ├── Second_floor
│   │   │   └── Top_of_the_tower
│   └── Observatory
│       └── star_chart
├── Forest
├── Garden
│   ├── Flower_garden
│   │   └── flowers
│   ├── Maze
│   └── Shed
│       ├── bucket
│       └── wheelbarrow
├── Mountain
│   ├── Cave
│   └── cauldron
└── Stall
```

Per una migliore comprensione della mappa di gioco così della navigazione a questo punto ho preferito fare un **tree**, è un comando che visualizza il contenuto di una directory sotto forma di struttura ad albero gerarchica, mostrando le sottodirectory e i file in modo nidificato.

È particolarmente utile per avere una rapida visione d'insieme della struttura di directory e file su più livelli.

Livello 3: Raggiungere la sala del trono

Obiettivo della missione:

```
Mission goal
=====

Go back to the starting location and then go to the throne room
using only two commands.
```

Raggiungere la sala del trono all'interno del castello.

Procedimento:

```
[mission 3] $ cd Castle/Main_building/Throne_room

[use 'gsh help' to get a list of available commands]
[mission 3] $ gsh check

Congratulations, mission 3 has been successfully completed!
```

Analisi:

In questo livello ho applicato una tecnica di navigazione più efficiente: invece di eseguire comandi `cd` sequenziali, ho concatenato il percorso completo usando il separatore `/`. Questo approccio è particolarmente utile quando si conosce il percorso esatto della directory di destinazione, riducendo il numero di comandi necessari.

Livello 4: Creazione directories `mkdir`

Obiettivo della missione:

Mission goal

Build a "Hut" in the forest, and then build a "Chest" in the hut.

Costruire una directory "Hut" nella foresta, e successivamente costruire una "Chest" all'interno della capanna.

Procedimento:

```
[mission 4] $ cd 06 ls
Castle Forest Garden Mountain Stall

~
[mission 4] $ cd 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!
```

Analisi:

Mi sono recato nella foresta e ho utilizzato il comando `mkdir` (make directory) per creare le strutture richieste: Il comando `mkdir` consente di creare nuove directory. In questo caso:

- Ho creato la directory "Hut" all'interno di Forest
- Ho navigato all'interno di Hut con `cd Hut`
- Ho creato la directory "Chest" all'interno di Hut

Questo livello introduce il concetto di creazione di strutture gerarchiche nel filesystem.

Livello 5: Rimozione di files con `rm`

Obiettivo della missione:

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.

Tornare alla cantina del castello e eliminare tutti i ragni, lasciando intatti i pipistrelli (in quanto presenti nello stemma del castello e ritenuti portatori di fortuna).

Procedimento:

```
[mission 5] $ cd && cd Castle/Cellar && ls
barrel_of_apples  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!
```

Analisi:

Ho navigato alla cantina e utilizzato il comando `rm` (remove) per eliminare selettivamente i ragni: il comando `rm` è uno dei più importanti (e pericolosi) in un ambiente Unix/Linux poiché consente l'eliminazione permanente di file e directory. In questo caso:

- Ho identificato i file da eliminare tramite `ls`
- Ho rimosso solo i file nominati "spider_*
- Ho lasciato intatti i file "bat_*" e "barrel_of_apples"

Questa missione enfatizza l'importanza della precisione quando si utilizzano comandi distruttivi come `rm`, poiché l'eliminazione è immediata e irreversibile nella maggior parte dei sistemi Unix/Linux.

Livello 6: Spostare files con `mv`

Obiettivo della missione:

Mission goal

Collect all the coins that you can find in the garden in front of the castle, and put them in your chest in your hut in the forest.

Cercare i coins nel giardino davanti al castello e spostarli al sicuro nella nostra cesta.

Procedimento:

```
~/Castle/Cellar
[mission 6] $ cd && cd Garden

~/Garden
[mission 6] $ ls
coin_1  coin_2  coin_3  Flower_garden  Maze  Shed

~/Garden
[mission 6] $ mv coin_1 coin_2 coin_3 ~/Forest/Hut/Chest

~/Garden
[mission 6] $ gsh check

Congratulations, mission 6 has been successfully completed!
```

Analisi:

Mi sono recato nel giardino e con un `ls` ho individuato i coins, quindi li ho spostati nella mia cesta selezionandoli con `mv coin_1 ... coin_n` a `~/Forest/Hut/Chest`.

Il comando `mv` è utilizzato per spostare o rinominare files e directories. In questo livello:

- Ho identificato tutti i files delle monete tramite `ls`
 - Ho utilizzato il percorso assoluto per spostare i file da `Garden` a `Forest/Hut/Chest`
 - Ho eseguito spostamenti multipli utilizzando una singola riga
-

Livello 7: TAB, `ls -A` e `mv` per trovare e raccogliere i coins

Obiettivo della missione:

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.

Raccogliere tutte le monete nel giardino davanti al castello e metterle nella cesta. Comprendere l'utilizzo del flag `-A` con `ls` per visualizzare file nascosti.

Impara “tabbare” per auto-completare i nomi di file o directory.

Procedimento:

```
~/Garden
[mission 7] $ ls -A
.13806_coin_1  .5697_coin_2  Maze
.49299_coin_3  Flower_garden  Shed

~/Garden
[mission 7] $ mv
.13806_coin_1  .5697_coin_2  Maze/
.49299_coin_3  Flower_garden/  Shed/

~/Garden
[mission 7] $ mv .
./              .13806_coin_1  .5697_coin_2
../             .49299_coin_3

~/Garden
[mission 7] $ mv .
./              .13806_coin_1  .5697_coin_2
../             .49299_coin_3

~/Garden
[mission 7] $ mv .13806_coin_1 .5697_coin_2 .49299_coin_3 ~/Forest/Hut/Chest

~/Garden
[mission 7] $ gsh check

Congratulations, mission 7 has been successfully completed!
```

Analisi:

Ho navigato nel giardino utilizzando `ls -A` per visualizzare tutti i file inclusi quelli nascosti, successivamente ho spostato le monete nella cesta utilizzando `mv`.

Questo livello approfondisce l'utilizzo del comando `ls` introducendo il flag `-A`, che visualizza tutti i file tranne "." e "..".

L'utilizzo di percorsi concatenati nel comando `mv` (es. `mv./*.coins ~/Forest/Hut/Chest`) dimostra l'efficienza di specificare il percorso completo in un'unica operazione, riducendo il numero di comandi necessari.

Livello 8: WildCards con `rm`

Obiettivo della missione:

Mission goal

Get rid of all the spiders that are crawling in the cellar.
Again, do not do not disturb the bats.

*

The "*" character stands in for any sequence of characters (including an empty sequence).

?

The "?" character stands in for any single character.

Those wildcards can be used to denote lists of existing files / directories in the current working directory.

Eliminare tutti i ragni nascosti nella cantina senza disturbare i pipistrelli.

Procedimento:

```
~/Castle/Cellar
[mission 8] $ ls
10215_spider_5  18557_spider_18  25173_spider_25  32279_spider_27
10298_spider_43 18941_spider_33  25851_spider_14  32620_spider_8
11637_spider_36 19416_spider_6   27378_spider_29  32760_spider_1
11847_spider_23 19832_spider_2   2740_spider_11   332_spider_44
11886_spider_26 2011_bat_1       27691_spider_15  4071_spider_47
12295_spider_7  20340_spider_3   28039_spider_12  4869_spider_35
13398_spider_40 20845_spider_4   28128_spider_41  5521_spider_19
13484_spider_16 21573_spider_37  29207_bat_4      587_spider_34
15015_spider_30 21631_bat_3      29571_spider_17  634_spider_42
15025_spider_45 22188_spider_28  29753_spider_10  6633_bat_5
1561_spider_46  22445_spider_21  29935_spider_38  8172_spider_32
16041_spider_39 23209_spider_48  31079_spider_13  8366_spider_24
1758_spider_20  23806_bat_2      31237_spider_22  8616_spider_31
18275_spider_50 24029_spider_9   313_spider_49    barrel_of_apples

~/Castle/Cellar
[mission 8] $ rm *spider*

~/Castle/Cellar
[mission 8] $ gsh check

Congratulations, mission 8 has been successfully completed!
```

Analisi:

Ho navigato nella cantina e utilizzato il comando `rm *spider*` per eliminare selettivamente tutti i file contenenti "spider" nel loro nome, preservando gli altri file come i pipistrelli e gli oggetti vari.

Questo livello introduce il concetto di wildcard in combinazione con `rm`:

- Il carattere `*` funge da pattern matching, quindi accetta al suo posto qualsiasi stringa
 - il carattere `?` accetta al suo posto un singolo carattere qualsiasi
-

Livello 9: Eliminazione di file nascosti

Obiettivo della missione:

Mission goal

The spiders are getting clever: they found a way to hide.
Get rid of all the spiders that are hiding in the cellar without disturbing the bats.

Eliminare tutti i ragni che strisciano nella cantina, che si nascondono usando file nascosti (nomi iniziati con punto).

Procedimento:

```
~/Castle/Cellar
[mission 9] $ ls .*spider*
.11192_spider_1  .21243_spider_34  .3877_spider_4
.12819_spider_37  .21495_spider_5  .4318_spider_43
.13048_spider_46  .214_spider_11   .4545_spider_48
.13138_spider_32  .21679_spider_49  .4864_spider_27
.13670_spider_18  .21925_spider_33  .491_spider_13
.14178_spider_41  .22062_spider_26  .5869_spider_20
.14644_spider_50  .22988_spider_9   .5999_spider_35
.15574_spider_44  .23221_spider_42  .6099_spider_3
.15866_spider_38  .24677_spider_6   .7280_spider_15
.16760_spider_14  .2510_spider_29   .7332_spider_39
.17212_spider_16  .26099_spider_45  .7337_spider_40
.17370_spider_2   .28352_spider_19  .7703_spider_30
.17434_spider_25  .28560_spider_12  .8134_spider_28
.17896_spider_23  .297_spider_31    .8274_spider_47
.17961_spider_7   .29899_spider_21  .9471_spider_10
.19628_spider_22  .3017_spider_36   .9855_spider_8
.2114_spider_24   .30794_spider_17

~/Castle/Cellar
[mission 9] $ rm .*spider*

~/Castle/Cellar
[mission 9] $ gsh check

Congratulations, mission 9 has been successfully completed!
```

Analisi:

Ho navigato nella cantina e utilizzato il comando `ls .*spider*` per visualizzare i ragni (testing inutile) `rm .*spider*` per eliminare tutti i file nascosti che contenevano ragni.

Questo livello incita l'uso dei wildcard e rimostra il concetto di file nascosti.

Livello 10: Utilizzo di `cp` per la copia selettiva

Obiettivo della missione:

Mission goal

You have taken a fancy to the four standards in the great hall of the castle. As stealing them would not go unnoticed, put a copy (same name, same content) of each in your chest.

Aggiungere alla nostra cesta una copia per ogni stendardo nella grande sala del castello

Procedimento:

```
[mission 10] $ ls
Castle/  Forest/  Garden/  Mountain/  Stall/

~
[mission 10] $ cd Castle/

~/Castle
[mission 10] $ ls
Cellar/  Great_hall/  Main_building/  Main_tower/  Observatory/

~/Castle
[mission 10] $ cd Great_hall/

~/Castle/Great_hall
[mission 10] $ ls -A
2061_stag_head      52876_decorative_shield  standard_2  standard_4
26300_suit_of_armour  standard_1              standard_3

~/Castle/Great_hall
[mission 10] $ cp standard_* ~/Forest/Hut/Chest

~/Castle/Great_hall
[mission 10] $ gsh check

Congratulations, mission 10 has been successfully completed!
```

Analisi:

Mi sono diretto nella sala grande del castello e ho utilizzato il comando `cp` (copy) per copiare selettivamente tutti i file degli standardi dentro la mia cesta. In questo livello:

- Ho identificato i quattro file degli standardi tramite `ls -A` nella Great_hall
- Ho utilizzato `cp` per copiare ogni standardo verso la destinazione finale nel forziere
- Diversamente da `mv`, il comando `cp` lascia il file originale intatto creando una copia

Questo livello introduce il concetto fondamentale della differenza tra spostamento e copia: `mv` modifica la struttura del filesystem spostando il file, mentre `cp` crea un clone mantenendo l'originale.

Livello 11: Un altro utilizzo di `cp`

Obiettivo della missione:

Mission goal

The tapestries in the castle's great hall are also particularly beautiful. Put a copy of each in your chest.

Crea una copia della tappezzeria, nella grande sala, dentro la tua cesta

Procedimento:

```
~/Castle/Great_hall
[mission 11] $ cp *tapestry* ~/Forest/Hut/Chest

~/Castle/Great_hall
[mission 11] $ gsh check

Congratulations, mission 11 has been successfully completed!
```

Analisi:

Evidentemente la missione 10 non era abbastanza e dovevamo praticare ancora le stesse tecniche di copiaggio file.

Apparte gli scherzi, come unica differenza qui c'è che abbiamo usato in combinazione la wildcard `*` che fa da pattern, invece che `?` per caratteri singoli.

Livello 12: `ls -l` per vedere la data di creazione, `cat` per il contenuto

Obiettivo della missione:

Mission goal

While wandering around the first floor of the main tower, some magnificent paintings catch your eye. Add a copy of the oldest one to your chest.

Secondary objectives

Take a moment to admire the sheer beauty of the paintings.

Copiare il file più antico tra i quadri del primo piano nella cesta.

Procedimento:

[illegible]


```
[mission 12] $ ls -l
total 16
-rw-rw-r-- 1 kali kali 1055 Sep 16 2016 painting_NLUFasGs
-rw-rw-r-- 1 kali kali 1454 Nov 4 1999 painting_PHKsRYfB
-rw-rw-r-- 1 kali kali 1502 Jan 26 1987 painting_VproxrSu
drwxrwxr-x 3 kali kali 4096 Nov 15 13:31 Second_floor/

~/Castle/Main_tower/First_floor
[mission 12] $ cp painting_VproxrSu ~/Forest/Hut/Chest

~/Castle/Main_tower/First_floor
[mission 12] $ gsh check

Congratulations, mission 12 has been successfully completed!
```

Analisi:

Ho usato **cat painting_name** per ammirare i quadri, poi **ls -l** per vederne la data di creazione, in fine **cp painting_name ~/Forest/Hut/Chest**

cat è semplice e stampa il contenuto del file chiamato.

ls -l produce un elenco esteso, una linea per ogni file, indicando da sinistra a destra:

1. i permessi
 2. il numero di collegamenti fisici al particolare file o directory
 3. il proprietario
 4. il gruppo assegnato
 5. le dimensioni apparenti in byte
 6. la data di ultima modifica in forma compatta
 7. il nome
-

Livello 13: `cal` per il calendario

Obiettivo della missione:

```
Mission goal
=====

Nostradamus predicted a spectacular star conjunction on the
07-27-2003.
But what will the day of the week be on that date?

When you have it, run the command ``gsh check``.
```

Scopri di che giorno casca la previsione di Nostradamus.

Procedimento:

```
~/Castle/Main_tower/First_floor
[mission 13] $ cal 07 2003
      July 2003
Su Mo Tu We Th Fr Sa
                1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

~/Castle/Main_tower/First_floor
[mission 13] $ gsh check
What was the day of the week for the 07-27-2003?
 1 : Monday
 2 : Tuesday
 3 : Wednesday
 4 : Thursday
 5 : Friday
 6 : Saturday
 7 : Sunday
Your answer: 7

Congratulations, mission 13 has been successfully completed!
```

Analisi:

Uso `cal 07 2003` per vedere il giorno della settimana di cui capiterà il 27 per prevedere la congiunzione di Nostradamus, poi seleziono Domenica nel `gsh check`.

Il comando `cal` serve per visualizzare un calendario direttamente dal terminale. Se lo lanciamo senza argomenti, mostra il calendario del mese corrente; possiamo indicare anche un mese (numerico) e un anno, ad esempio `cal 07 2025` per vedere luglio 2025. È molto utile per orientarsi rapidamente sulle date, fare controlli su particolarità come giorni della settimana o semplicemente organizzare attività a terminale senza aprire altre applicazioni.

Livello 14: Test di `alias` per poter creare shortcuts

Obiettivo della missione:

Mission goal

Checking for hidden files is taking too long!

Create an alias "la" to run the command ``ls -A`` in order to list all files, including hidden ones, with only 2 letters.

Define the synonym

la

for the command

ls -A

and check that it works as expected.

How fortunate, there is a nice rock hidden just where you are.

Creare un alias `la` che esegua il comando `ls -A` per visualizzare rapidamente tutti i file, inclusi quelli nascosti, con due soli caratteri.

Procedimento:

```
~/Castle/Main_tower/First_floor
[mission 14] $ alias la='ls -A'

~/Castle/Main_tower/First_floor
[mission 14] $ la
.nice_rock  painting_NLUFasGs  painting_PHKsRYfB  painting_VproxrSu  Second_floor/

~/Castle/Main_tower/First_floor
[mission 14] $ gsh check

Congratulations, mission 14 has been successfully completed!
```

Analisi:

Il comando `alias` permette di definire abbreviazioni personalizzate per i comandi. In questo livello si crea `la` come alias di `ls -A`, così da velocizzare la chiamata. L'alias creato nella sessione corrente resta attivo fino alla chiusura della shell; per renderlo permanente occorre aggiungerlo al file di configurazione (`.bashrc` o simili).

Livello 15: Il Text Editor nano

Obiettivo della missione:

```
Mission goal
=====

Create a file named "journal.txt" in your chest and write a short
message in it.
You can use this file to record your notes and solutions for the
upcoming missions.

Details
-----

``nano`` is a command-line text editor. You can use it whenever
you need to edit a file from the shell.
```

Creare un file chiamato "journal.txt" nella Chest e scrivere al suo interno un breve messaggio.

Procedimento:

```
~/Castle/Main_tower/First_floor
[mission 15] $ cd

~
[mission 15] $ cd Forest/Hut/Chest

~/Forest/Hut/Chest
[mission 15] $ nano journal.txt

~/Forest/Hut/Chest
[mission 15] $ gsh check

Congratulations, mission 15 has been successfully completed!
```

Analisi:

Mi sono spostato nella directory Forest/Hut/Chest e ho usato il comando `nano journal.txt` per creare il file e scrivere il messaggio richiesto.

`nano file.txt`, è un editor di testo da terminale, permette di creare e modificare file testuali direttamente nella shell, senza uscire dall'ambiente. Se file.txt esiste lo apre altrimenti lo crea.

Livello 16: alias per aprire con nano il journal.txt

Obiettivo della missione:

```
Mission goal
=====

Create an alias "journal" in order to easily edit your journal file
wherever you are.

Details
=====

To edit the journal file with ``nano`` from, for example, the cellar
or the throne room, you need to give the full path to the file:
"~/Forest/.../journal.txt".

To avoid typing this long command each time, you can create an alias
just like

alias la='ls -a'
```

Creare un alias **journal** che apra con **nano** il file **journal.txt** nella cesta usando il percorso completo.

Procedimento:

```
~/Forest/Hut/Chest
[mission 16] $ alias journal="nano ~/Forest/Hut/Chest/journal.txt"

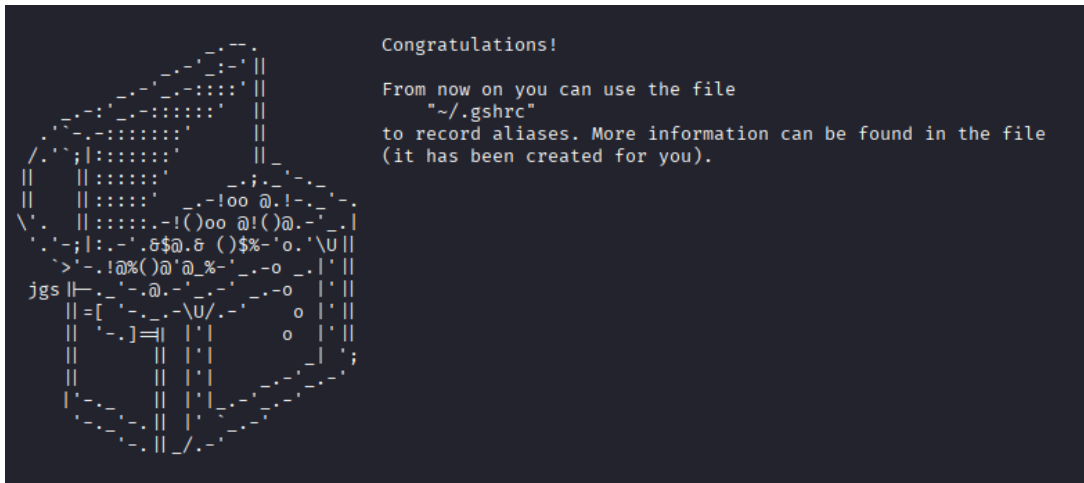
~/Forest/Hut/Chest
[mission 16] $ gsh check

Congratulations, mission 16 has been successfully completed!
```

Analisi:

Il comando **alias** in questa missione viene usato per velocizzare l'accesso al file **journal.txt**, evitando di scrivere ogni volta il percorso lungo. L'uso del percorso assoluto (~ indica la home) garantisce che il file sia richiamabile dalla shell ovunque ci si trovi. Questa tecnica è utile per creare scorciatoie operative personalizzate quando serve accedere sporadicamente o frequentemente agli stessi file, riducendo i tempi di digitazione.

Sbloccate Alias Permanenti



Adesso è possibile scrivere alias, riutilizzabili in più sessioni,
all'interno di `~/.gshrc` e quelli resteranno sempre in memoria

```
# This file is loaded (or "sourced" in shell jargon) when the game
# starts. You can use it to record aliases (and also other things)
# that you may want to keep across game sessions. Indeed, an alias
# defined in the shell will be "forgotten" if it is not added here
# (aliases defined as part of missions are recorded for you).
#
# Remark: all lines starting with '#' in this file are ignored.
# Standard shell use such files, usually hidden in the home
# directory. For bash, the file is found at the path
#     ~/.bashrc
# and the one for zsh at
#     ~/.zshrc
# (Do not modify those files as they are used by GameShell!)
#
# Whenever you edit this file, the definitions it contains are not
# directly added to the current shell. You can however rely on the
# following alias to do that for you.
alias reload="source ~/.gshrc"
#
# Here are a few alias ideas for you:
# - an alias "gg" for "gsh goal",
# - an alias "gc" for "gsh check",
# - an alias "editrc" to edit this file with nano.
```

Livello 17: WildCards ON Fire

Obiettivo della missione:

```
Mission goal
=====

At the back of the cellar, there is a small opening going to the
spider queen's lair.
Go there, and remove the spider queen (and nothing else).

Note: you have a limited amount of time (20 seconds) to do that.
You can use the command ``gsh reset`` to reset the timer.

Another thing: shell patterns have been deactivated. You cannot
use the wildcards ``*`` or ``?``.
```

Accedere alla tana della spider queen nella Cellar ed eliminare solo il file relativo alla spider queen, senza usare wildcard e con tempo limitato.

Procedimento:

```
[mission 17] $ cd .Lair_of_the_spider_queen\ kVORFYmAWRBInkdK xGzkkIhtdYvmaDnf/
~/Castle/Cellar/.Lair_of_the_spider_queen kVORFYmAWRBInkdK xGzkkIhtdYvmaDnf
[mission 17] $ ls -A
nOKLEovjnGCiOQBD_baby_bat_RlRQvcqfaKVjSqlW YzLIKaPmgCAwuNkA_spider_queen_EIUccKjPosORMMSn
~/Castle/Cellar/.Lair_of_the_spider_queen kVORFYmAWRBInkdK xGzkkIhtdYvmaDnf
[mission 17] $ rm YzLIKaPmgCAwuNkA_spider_queen_EIUccKjPosORMMSn
~/Castle/Cellar/.Lair_of_the_spider_queen kVORFYmAWRBInkdK xGzkkIhtdYvmaDnf
[mission 17] $ gsh check
Perfect, it took you only 15 seconds to complete this mission!
Congratulations, mission 17 has been successfully completed!
```

Analisi:

Mi sono spostato nella directory `.Lair_of_the_spider_queen`. Ho usato `ls -A` per elencare i file e individuare il nome esatto del file relativo alla spider queen, il flag `-A` permette di visualizzare anche i file che sarebbero nascosti.

Ho lanciato il comando `rm NomeDelFileSpiderQueen` tabbando il nome completo, e in fine `gsh check` in 15 secondi.

Livello 18: Mi Vedi???

Obiettivo della missione:

```
Mission goal
=====

As you are walking around the castle, you feel like you are being
watched... Turn your head quickly enough and you may see one of
the paintings' eyes following you.

1/ Run the ``xeyes`` command, and stop it.
2/ Run the ``xeyes`` command in the background.
```

Procedimento:

```
[mission 18] $ xeyes
^C

~
[mission 18] $ xeyes&
[1] 29748

~
[mission 18] $ gsh check
Congratulations, mission 18 has been successfully completed!
```

Analisi: apriamo `xeyes` e si chiude con ctrl+c, ma poi si riapre con `xeyes&` così che rimane eseguito in background, e ora che ho completato il gioco è ancora lì a guardarmi.....

Livello 19: & comandi in parallelo

Obiettivo della missione:

```
Mission goal
=====

The king's pyrotechnician appears next to you. He asks you to fire
**at least 3 consecutive fireworks** so he can see them from far
away.

A single firework can be created with the magical word

    flarigo
```

Procedimento:

```
[mission 19] $ flarigo & flarigo & flarigo & gsh check
```

[2] 33845

[3] 33846

[4] 33847

Let's have a look:

$$\begin{array}{c} \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ - : O : - \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot \end{array}$$

```
[2] Done
```

flarigo

[3]- Done

flarigo

[4]+ Done

flarigo

Great, that looked good!

Congratulations, mission 19 has been successfully completed!

Analisi:

Nella shell puoi separare diversi comandi con `&` che ne permette l'esecuzione in parallelo, questo esercizio era risolvibile solo lanciando tutti i fuochi d'artificio ed eseguendo il `gsh check` contemporaneamente, così che il pirotecnico riesca a vederli in diretta.

Livello 20: Spam charmiglio pipi

Obiettivo della missione:

Mission goal

The king's pyrotechnician is trying to remember the magical incantation for creating the grand finale for his fireworks. This incantation starts with the word charmiglio and must be followed by four random letters, as in

```
$ charmiglio abcd  
or  
$ charmiglio oops
```

Help the pyrotechnician by finding 4 letters producing appropriate fireworks.

NOTE: when the four letters are incorrect, the magical reaction can take a very long time. You need to interrupt it!

It will probably take several tries before finding a combination of letters that works.

Procedimento:

```
[mission 20] $ charmiglio pipi
.-):(*_#.#_
-#.*.)(:-.....
**%#--#*_%.)
..).(*:_#..(
((*_..)%_*#
#**:_-.#)*-)
%_-_#-*(###
)_**_#((.(._.*.
*:-(.#:-_*
#-._*(*.%%_
:._%#.*-_
):*._#___:_
:_%._%(. (
.*#-#.%#)%
._-###)-(*_).
:(.*.###-#-:%#
:;%_..-_*-_-)(
.%#)_..-)__.
.%)*)__%:::_#
.:#:#. *_.(
```

It works! The special incantation is `pipi`

[illegible]

```
[mission 20] $ gsh check
What's a valid 4 letters sequence? pipi

Congratulations, mission 20 has been successfully completed!
```

Analisi:

Spammare **charmiglio xxxx** e ctrl+c dopo 3 secondi circa finché non si trova la combinazione giusta, poi fai il **gsh check** con la combinazione trovata

Livello 21: File Hunting

Obiettivo della missione:

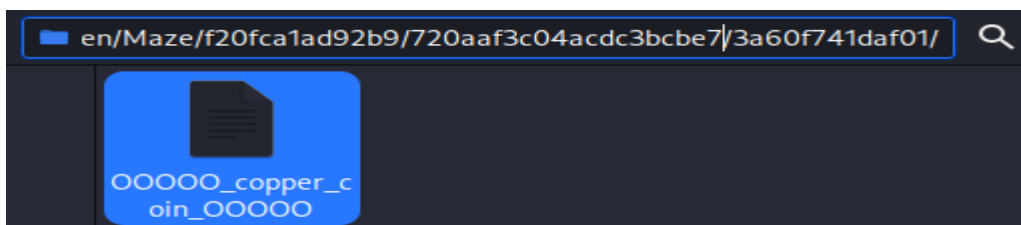
```
Mission goal
=====

Find the copper coin in the small maze in the garden and move it to
your chest.

If you want, you can use a graphical file manager. The maze is to be
found in directory
/home/kali/gameshell.1/World/Garden/Maze
```

Cerca il Conio di rame nel labirinto.

Procedimento:



```
[mission 21] $ cd Garden/Maze/f20fca1ad92b9/720aaf3c04acdc3bcbe7/3a60f741d
af01/

~/Garden/Maze/f20fca1ad92b9/720aaf3c04acdc3bcbe7/3a60f741daf01
[mission 21] $ ls
OOOOO_copper_coin_OOOOO

~/Garden/Maze/f20fca1ad92b9/720aaf3c04acdc3bcbe7/3a60f741daf01
[mission 21] $ mv OOOOO_copper_coin_OOOOO ~/Forest/Hut/Chest

~/Garden/Maze/f20fca1ad92b9/720aaf3c04acdc3bcbe7/3a60f741daf01
[mission 21] $ gsh check

Congratulations, mission 21 has been successfully completed!
```

Analisi:

Come venivano cercati i file nella preistoria, ogni directory una ad una come un binary sort vivente, finché non compariva il `OOOOO_copper_coin_OOOOO`

Livello 22: File Hunting con tree

Obiettivo della missione:

Mission goal

Find the silver coin in the maze in the garden and move it to your chest using the shell.

Ora cerca il conio argentato, sempre nel labirinto

Procedimento:

```
[mission 22] $ tree
.
├── 14df7c773f7e93ec
│   ├── 61629ab486df5023648782
│   │   ├── 5a551fb05637f3cbc9cddec3a
│   │   ├── 87d1763fff93688175a071ab6
│   │   └── f5eea7882d45564f3824023e67c92a
│   ├── 6a85bda272c1440184
│   │   ├── 5250527a1bdb1c4226b
│   │   ├── 777fe0c79c4fd234c
│   │   ├── 98b7216c96bd6fd2abb52a315b1675
│   │   └── e9a8c822a074d7d83cfa8d0bf6581be
│   │       ├── 205e002c7212f
│   │       ├── 6cc2e504cc17437df
│   │       └── ac24c3d6535733ebc17189b632d87b
│   │           └── 00000_silver_coin_00000
├── 6b98953348fb4
│   ├── 015353958211ec
│   │   ├── 0627e6fa541c27a55cbc8dd7d425ad
│   │   ├── 4f7a1d9dd84d5d655f3
│   │   └── 911378201e39f701ac686442f30
│   ├── 3cb339c1541c622f3
│   │   ├── 096b3480e3359761796ae
│   │   ├── d7df0891f29f7
│   │   └── ee8bde31aa207
│   ├── 74a634c4d696012e91ee05
│   │   ├── 23d0ffc671f28b3dd192261f7
│   │   ├── 80f904f038
│   │   └── a53c2044fb3052
└── dfb9434fc891
    ├── 0120e752765c6ebda8f4b102f42fbc
    │   ├── 0163434835feb73857a2540
    │   ├── 10f10de64bd94bcc265b48165cf018
    │   └── 9d31ac94fe5a4f9ed0
    ├── c50e0eff2b0d5
    │   ├── 837a151dbcd
    │   ├── 934a74d3fa98b246d3bee9ea3e919d
    │   ├── b50053c1f829cb21985e6f32c8ec
    │   └── ceee94b74954e616a9
    ├── 51dc035331
    │   ├── cea0dc68d0bf698519c
    │   └── f4d9a050a
```

```
~/Garden/Maze
[mission 22] $ cd 14df7c773f7e93ec/e9a8c822a074d7d83cfa8d0bf6581be/ac24c3d
6535733ebc17189b632d87b

~/Garden/Maze/14df7c773f7e93ec/e9a8c822a074d7d83cfa8d0bf6581be/ac24c3d6535
733ebc17189b632d87b
[mission 22] $ mv 00000_silver_coin_00000 ~/Forest/Hut/Chest

~/Garden/Maze/14df7c773f7e93ec/e9a8c822a074d7d83cfa8d0bf6581be/ac24c3d6535
733ebc17189b632d87b
[mission 22] $ gsh check

Congratulations, mission 22 has been successfully completed!
```

Analisi:

Grazie al `tree` Si può dichiarare finita l'era del rame... facilita l'orientamento e la ricerca di file in modo esponenziale, è un comando tra i più semplici ma più potenti allo stesso tempo.

Livello 23: File Hunting con `find` e WildCards

Obiettivo della missione:

Mission goal

Find the gold coins in the maze hidden in the garden and move them to your chest.

L'ora del conio dorato è arrivata, devi trovarlo a tutti i costi.

Procedimento:

```
~/Garden/Maze
[mission 23] $ find -iname "*gold*"
./8858a26cb2cc/361182735bfa00127d3057121bd08/c4fce47f207cc8/Gold_Coin_2
./905357bfd9a1272a953d/143f7053f33064a2d/76cc7650d8fc/gold_coin_1

~/Garden/Maze
[mission 23] $ ^C

~/Garden/Maze
[mission 23] $ mv ./8858a26cb2cc/361182735bfa00127d3057121bd08/c4fce47f207cc8/Gold_Coin_2 ./905357bfd9a1272a953d/143f7053f33064a2d/76cc7650d8fc/gold_coin_1 ~/Forest/Hut/Chest

~/Garden/Maze
[mission 23] $ gsh check

Congratulations, mission 23 has been successfully completed!
```

Analisi: Find e WildCards

`find [percorso] [criteri] [azione]`

`find` è uno strumento potentissimo per cercare file e directory nel filesystem ricorsivamente, partendo da una directory specificata. È estremamente flessibile grazie ai molteplici criteri di ricerca e azioni eseguibili su ogni oggetto trovato.

Le wildcards sono caratteri speciali utilizzati per rappresentare uno o più caratteri arbitrari in una stringa o pattern. Sono fondamentali nelle shell Unix/Linux, pattern matching, ricerca file, filtri e scripting per definire criteri di selezione flessibili senza conoscere esattamente il nome completo, esempi più usati:

- `*` Corrisponde a zero o più caratteri qualsiasi, inclusi vuoti.
 - `?` Corrisponde esattamente a un carattere qualsiasi singolo.
 - `[abc]` Corrisponde a uno qualsiasi dei caratteri tra parentesi.
 - `[a-z]` o `[0-9]` Corrispondono a tutti i valori del range.
-

Livello 24: Herbal Tea per imparare head

Obiettivo della missione:

Mission goal

A forgetful old hermit called Servillus has set up camp in a cave with his old, leather-bound potion book.
Go to the cave and help him remember the recipe of his famous herbal tea.

In order to validate the mission, you need to be in the cave with Servillus **and** your last command prior to ``gsh check`` must show the recipe (including its title), but nothing else.

Note: you shouldn't alter the content of the book of potions.

Procedimento:

```
~/Mountain/Cave/Book_of_potions
[mission 28] $ cat table_of_contents
1. Transformation potion _____ pages 1-2
2. Elixir of youth _____ pages 3-4
3. Philter of love _____ page 5
4. Bottled death (powerful poison) _____ page 6
5. Herbal tea _____ page 7
6. Draft of invisibility _____ pages 7-8
7. Homeopathic healing potion (part 1) _____ pages 8-9
8. Homeopathic healing potion (part 2) _____ page 10
9. Homeopathic healing potion (part 3) _____ page 11
10. Toadstool stew _____ page 12
11. Distilled water _____ page 13
12. King's ale _____ Page 13
```

```
[mission 24] $ head -n 6 Book_of_potions/page_07
vvvvvvvvvv
Herbal tea
^^^^^^^^^^
1) Boil water.
2) Add herbs from the forest.
3) Let it sit for five minutes and drink while hot.
```

```
~/Mountain/Cave
[mission 24] $ gsh check
```

Congratulations, mission 24 has been successfully completed!

Analisi:

`cat Book_of_potions/table_of_contents` stampa l'intero indice, leggendolo trovo subito le linee che mi interessano: la ricetta per l'**Herbal tea** a pagina 7. In fine quindi per rispettare la consegna scriviamo solo le righe inerenti alla ricetta (titolo incluso) tralasciando le restanti della pagina 7:

`head -n 6 Book_of_potions/page_07` che stampa esattamente le prime 6 righe.

Livello 25: Toadstool Stew per imparare `tail`

Obiettivo della missione:

Mission goal

The old man seems to enjoy your company very much. He invites you to stay for supper, and starts preparing a delicious stew for the both of you. While getting the cauldron ready he asks for your help. Read him the steps of the recipe from his book.

In order to validate the mission, you need to be in the cave with Servillus **and** last command prior to `gsh check` must show the steps of the recipe (without its title).

Note: you shouldn't alter the content of the book of potions.

Procedimento:

```
~/Mountain/Cave/Book_of_potions
[mission 28] $ cat table_of_contents
1. Transformation potion _____ pages 1-2
2. Elixir of youth _____ pages 3-4
3. Philter of love _____ page 5
4. Bottled death (powerful poison) _____ page 6
5. Herbal tea _____ page 7
6. Draft of invisibility _____ pages 7-8
7. Homeopathic healing potion (part 1) _____ pages 8-9
8. Homeopathic healing potion (part 2) _____ page 10
9. Homeopathic healing potion (part 3) _____ page 11
10. Toadstool stew _____ page 12
11. Distilled water _____ page 13
12. King's ale _____ Page 13
```

```
~/Mountain/Cave
[mission 25] $ tail -n 9 Book_of_potions/page_12
1) Boil water in a cauldron.
2) Add in a few death caps (Amanita phalloides).
3) Also add a few fly agarics (Amanita muscaria).
4) And some destroying angels (Amanita virosa).
5) Mix in a few deadly webcaps (Cortinarius rubellus).
6) Feel free to add in any colourful fungi you have on hand.
7) Let half of the water evaporate.
8) Season with a pinch of salt and a few herbs.
9) Serve hot in a bowl.

~/Mountain/Cave
[mission 25] $ gsh check

Congratulations, mission 25 has been successfully completed!
```

Analisi:

`cat Book_of_potions/table_of_contents`: trovo che la **Toadstool Stew** sta a pagina 12, e devo stampare solo ed esclusivamente la sua ricetta, nient'altro prima o dopo, quindi occorrerà rimuovere le prime 3 righe di titolo, equivalente a stampare solo le ultime 9 righe, che sono il numero dei passaggi precisi:

`tail -n 9 Book_of_potions/page_12`: stampa 9 righe del file, partendo dal fondo, l'esatto opposto del comando visto prima **head**

Livello 26: Transformation Potion per concatenare il `cat` di più files

Obiettivo della missione:

Mission goal

While cleaning the dishes, Servillus mentions an interesting potion that lets the drinker (temporarily) take the physical appearance of anyone.

Read the recipe of the potion from the hermit's book.

In order to validate the mission, you need to be in the cave with Servillus **and** your last command prior to `gsh check` must show the whole recipe (with its title).

Note: you shouldn't alter the content of the book of potions.

Procedimento:

```
~/Mountain/Cave/Book_of_potions
[mission 28] $ cat table_of_contents
1. Transformation potion _____ pages 1-2
2. Elixir of youth _____ pages 3-4
3. Philter of love _____ page 5
4. Bottled death (powerful poison) _____ page 6
5. Herbal tea _____ page 7
6. Draft of invisibility _____ pages 7-8
7. Homeopathic healing potion (part 1) _____ pages 8-9
8. Homeopathic healing potion (part 2) _____ page 10
9. Homeopathic healing potion (part 3) _____ page 11
10. Toadstool stew _____ page 12
11. Distilled water _____ page 13
12. King's ale _____ Page 13
```

```
[mission 26] $ cat Book_of_potions/page_01 Book_of_potions/page_02
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
Transformation potion
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
1) Boil water in a cauldron.
2) Add 3 measures of fluxweed to the cauldron.
3) Add 2 bundles of knotgrass to the cauldron.
4) Stir 4 times, clockwise.
5) Wave your wand then let potion brew for 80 minutes.
6) Add 4 leeches to the cauldron.
7) Crush 2 scoops of lacewing flies to a fine paste.
8) Add 2 measures of the crushed lacewings to the cauldron.
9) Heat for 30 seconds on a low heat.
10) Add 3 measures of boomslang skin to the cauldron.
11) Crush a bicorn horn into a fine powder.
12) Add 1 measure of the crushed horn to the cauldron.
13) Heat for 20 seconds at a high temperature.
14) Wave your wand then let potion brew for 24 hours.
15) Add 1 additional scoop of lacewings to the cauldron.
16) Stir 3 times, counter-clockwise.
17) Split potion into multiple doses, if desired.
18) Add a pieces of the person you wish to become.
19) Wave your wand to complete the potion.
```

```
~/Mountain/Cave
[mission 26] $ gsh check
```

Congratulations, mission 26 has been successfully completed!

Analisi:

`cat Book_of_potions/table_of_contents`: trovo che la `Transformation Potion` sta a pagine 1-2, e devo stampare sia titolo che ricetta sta volta per poter passare il livello, quindi:

`cat Book_of_potions/page_01 Book_of_potions/page_02`: titolo e ricetta stampati, in questo caso passando più file al `cat` gli concatena nell'output come si può vedere dalla seconda foto.

Livello 27: Elixir of Youth per scoprire **I**, come dare flow ai dati

Obiettivo della missione:

Mission goal

The old hermit notices your interest for potion recipes, and sees promise in your ability to lookup lists of ingredients. He challenges you to find the steps for the elixir of Youth.

In order to validate the mission, you need to be in the cave with Servillus **and** your last command prior to ``gsh check`` must show the steps for the recipe and nothing else.

Note: you shouldn't alter the content of the book of potions.

Procedimento:

```
~/Mountain/Cave/Book_of_potions
[mission 28] $ cat table_of_contents
1. Transformation potion _____ pages 1-2
2. Elixir of youth _____ pages 3-4
3. Philter of love _____ page 5
4. Bottled death (powerful poison) _____ page 6
5. Herbal tea _____ page 7
6. Draft of invisibility _____ pages 7-8
7. Homeopathic healing potion (part 1) _____ pages 8-9
8. Homeopathic healing potion (part 2) _____ page 10
9. Homeopathic healing potion (part 3) _____ page 11
10. Toadstool stew _____ page 12
11. Distilled water _____ page 13
12. King's ale _____ Page 13
```

```
~/Mountain/Cave
[mission 27] $ cat Book_of_potions/page_03 Book_of_potions/page_04 | ta
il -n 16
1) Fill a cauldron with used bath water.
2) Put a moderately large frog in the water.
3) Let the preparation rest overnight.
4) The next morning thank and free your little green friend.
5) Boil the water and add in a few sticks of oak tree.
6) Crush 5 river stones to a fine powder.
7) Mix in a third of the powder and stir vigorously.
8) Let the preparation rest for a day.
9) Add hairs from the tail of a squirrel (willingly given).
10) Add the remaining stone powder.
11) Stir the potion very vigorously, in all directions.
12) Take some time to rest after such an effort.
13) Rest a little bit more.
14) Even take a nap if you want.
15) Add a few larch tree needles for seasoning.
16) Drink the potion from the cauldron.

~/Mountain/Cave
[mission 27] $ gsh check
Congratulations, mission 27 has been successfully completed!
```

Analisi:

`cat Book_of_potions/table_of_contents`: trovo che la `Elixir of Youth` sta a pagine 3-4, e devo stampare solo la ricetta sta volta per poter passare il livello, quindi sapendo che sono 16 passaggi totali:

```
cat Book_of_potions/page_03 Book_of_potions/page_04 | tail -n 16
```

Quel carattere si chiama pipe: `|`, ed è veramente utile nella shell, perché ti permette di eseguire un primo comando come l'esempio sotto (`command1`) e solo una volta terminata la sua esecuzione il suo output viene passato come parametro d'ingresso per il secondo comando e viene eseguito (`command2`):

```
command1 | command2
```

Livello 28: Distilled Water per testare ogni skill appresa

Obiettivo della missione:

Mission goal

The old hermit is thirsty and he would like you lookup the recipe for distilled water.

In order to validate the mission, you need to be in the cave with Servillus ****and**** your last command prior to ``gsh check`` must show the steps for the recipe and nothing else.

Note: you shouldn't alter the content of the book of potions.

Procedimento:

```
~/Mountain/Cave/Book_of_potions
[mission 28] $ cat table_of_contents
1. Transformation potion _____ pages 1-2
2. Elixir of youth _____ pages 3-4
3. Philter of love _____ page 5
4. Bottled death (powerful poison) _____ page 6
5. Herbal tea _____ page 7
6. Draft of invisibility _____ pages 7-8
7. Homeopathic healing potion (part 1) _____ pages 8-9
8. Homeopathic healing potion (part 2) _____ page 10
9. Homeopathic healing potion (part 3) _____ page 11
10. Toadstool stew _____ page 12
11. Distilled water _____ page 13
12. King's ale _____ Page 13
```

```
~/Mountain/Cave
[mission 28] $ head -n 6 Book_of_potions/page_13 | tail -n 3
1) Boil water in a big pot.
2) Condense the vapor in a fresh container.
3) Add minerals for a better taste (optional).
```

```
~/Mountain/Cave
[mission 28] $ gsh check
```

Congratulations, mission 28 has been successfully completed!

Analisi:

`cat Book_of_potions/table_of_contents`: trovo che la `Distilled Water` sta a pagina 13 insieme al `King's Ale`, e sta volta devo stampare solo la ricetta per poter passare il livello quindi intanto isolerei la formula per la `Distilled Water`:

- `head -n 6 Book_of_potions/page_13`: e ora manca solo il titolo che si rimuove facilmente con:
- `tail -n 3`, non vuole un file questa volta perché useremo proprio l'output dato dall'`head`

Quindi `head -n 6 Book_of_potions/page_13 | tail -n 3` per la ricetta della `Distilled Water` senza titolo.

Livello 29: `ps` e `kill`, gestione processi

Obiettivo della missione:

Mission goal

A mischievous imp cast a spell that puts smudges of coal everywhere in the castle.
Find this spell and remove it.

Remark

The spell is a process.

Procedimento:

```
~
[mission 29] $ ps

      *#@*
      @_**/~
      !$-#

  PID TTY          TIME CMD
  31214 pts/2        00:00:00 zsh
  118706 pts/2        00:00:00 bash
  118760 pts/2        00:00:00 bash
  119119 pts/2        00:00:00 spell
  120483 pts/2        00:00:00 ps

~
[mission 29] $ kill 119119
[1]+  Terminated                  "$GSH_TMP/${gettext "spell"}"

~
[mission 29] $ kill 11911
bash: kill: (11911) - No such process

~
[mission 29] $ gsh check

Congratulations, mission 29 has been successfully completed!
```

Analisi:

Qui vediamo due comandi che sono strumenti fondamentali per il controllo e la gestione dei processi in ambiente Linux.

ps Visualizza un'istantanea dei processi attivi nel sistema al momento dell'esecuzione. Flag importanti sono:

- **-e o -A** Mostrano tutti i processi
- **-f** formato esteso, con PPID, USER, tempo CPU, ecc.
- **aux** elenca ogni processo che è esistito, e dice anche lo stato

kill Invia segnali a processi per controllarne il comportamento, principalmente per terminare un processo (infatti **kill**)

Livello 30: `kill -l` e i segnali

Obiettivo della missione:

Mission goal

The mischievous imp has more than one trick up his sleeve. He managed to protect his spell against most tampering. You need to find this spell and try to remove it with standard signal. If it doesn't work, use a more brutal signal.

Procedimento:

```
[mission 30] $ kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT      4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS      8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD    18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN    22) SIGTTOU    23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF    28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS     34) SIGRTMIN  35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5 60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

```
[mission 30] $ pkill -15 spell
You'll need to do better than that to kill my spell!
```

```
  *#@*
  5_**/~
  !$-#
```

```
  *#@*
  5_**/~
  !$-#
```

```
~
[mission 30] $ pkill -9 spell
```

```
  *#@*
  5_**/~
  !$-#
```

```
~
[mission 30] $ ps
  PID TTY          TIME CMD
 31214 pts/2      00:00:01 zsh
134985 pts/2      00:00:00 bash
135039 pts/2      00:00:00 bash
137063 pts/2      00:00:00 ps
```

```
~
[mission 30] $ gsh check
```

```
Congratulations, mission 30 has been successfully completed!
```

Analisi:

Qua ci metteva in difficoltà perché la spell non sarebbe morta semplicemente con `kill` che di default usa in segnale 15 (`SIGTERM`), quindi occorre prima provarci e poi trovare un segnale che lo uccidesse, allora provando successivamente con il segnale 9 (`SIGKILL`) ha funzionato, quest'ultimo uccide forzatamente il processo senza cleanup.

Livello 31: pstree fusione tra ps e tree

Obiettivo della missione:

```
Mission goal
=====

The imp is comparing his magic with a fairy. They met in the
cellar, and imp is conjuring lumps of coal while the fairy
is conjuring delicate snowflakes.

Remove the imp's spells and the coal that litters the
cellar, but don't touch the snowflakes!

Remark
=====

Do not kill the imp or the fairy.
```

Procedimento:

```
~
[mission 31] $ cd Castle/Cellar/

~/Castle/Cellar
[mission 31] $ pstree $$ -p
bash(135039)---mischievous_imp(264919)---spell(264935)---sleep(265908)
                                     |---spell(264936)---sleep(265993)
                                     |---spell(264937)---sh(266022)
                                     |---tail(264938)
      |---nice_fairy(264918)---spell(264925)---sleep(265913)
      |                      |---spell(264927)---sleep(265986)
      |                      |---spell(264928)
      |                      |---tail(264929)
      |---pstree(266024)

~/Castle/Cellar
[mission 31] $ kill 264935 264936 264937

~/Castle/Cellar
[mission 31] $ pstree $$ -p
bash(135039)---mischievous_imp(264919)---tail(264938)
      |---nice_fairy(264918)---spell(264925)---sleep(267832)
      |                      |---spell(264927)---sleep(267876)
      |                      |---spell(264928)---sleep(267788)
      |                      |---tail(264929)
      |---pstree(267877)

~/Castle/Cellar
[mission 31] $ rm *coal*

~/Castle/Cellar
[mission 31] $ gsh check

Congratulations, mission 31 has been successfully completed!
```

Analisi:

`ps tree` Visualizza i processi in esecuzione come un albero gerarchico, mostrando la relazione padre-figlio tra processi. Flag importanti:

- `-p` Mostra il PID accanto a ogni processo.
- `-u` Mostra l'utente proprietario di ogni processo.
- Specificando un PID o un nome utente limita l'albero ai processi corrispondenti.

Nell'esercizio una volta trovati i processi target poi è bastato un `kill` seguito dai singoli PID di ognuno, e in fine chiedeva di rimuovere il carbone lasciato dall'imp in Cellar con un semplice `rm *coal*` nella directory.

Livello 32: Gara di somme

Obiettivo della missione:

Mission goal

To get better in the magical art, one needs to know mental math.

Get ready, because Merlin is about to test your precision with sums.

Run the command `gsh check` to start.

Procedimento:

```
~/Castle/Cellar
[mission 32] $ gsh check
62 + 46 = ?? 108
38 + 55 = ?? 93
34 + 59 = ?? 93
85 + 90 = ?? 175
70 + 12 = ?? 82

Congratulations, mission 32 has been successfully completed!
```

Analisi:

Livello senza bisogno di spiegazioni, semplice gioco dove chiedeva di sommare 2 numeri fino al completamento del livello.

Livello 33: Mathematics_101 e <

Obiettivo della missione:

Mission goal

To get better in the magical art, one needs to know mental math.

Get ready, because Merlin is about to test your speed with products.

Run the command ``gsh check`` to start.

Remark

There now is a time constraint.

Hint

The library is rumored to contain some mathematics books and hidden volumes.

Procedimento:

```
Main_building
├── Library
│   ├── Greek_Latin_and_other_modern_languages
│   ├── Mathematics_101
│   └── Merlin_s_office
│       ├── candle
│       ├── Drawer
│       └── ink_and_scroll
```

```
~/Castle/Main_building/Library
[mission 33] $ gsh check < ~/Castle/Main_building/Library/Mathematics_101
31 * 2 = ?? 17 * 79 = ?? 66 * 19 = ?? 83 * 81 = ?? 62 * 66 = ?? 70 * 55 = ?? 70 * 12 = ??
31 * 58 = ?? 54 * 96 = ?? 13 * 20 = ?? 36 * 60 = ?? 66 * 42 = ?? 44 * 12 = ?? 83 * 12 = ??
?? 77 * 46 = ?? 44 * 76 = ?? 14 * 84 = ?? 10 * 86 = ?? 62 * 94 = ?? 57 * 58 = ?? 97 * 78
= ?? 98 * 19 = ?? 92 * 2 = ?? 95 * 5 = ?? 77 * 37 = ?? 21 * 41 = ?? 61 * 79 = ?? 70 * 67
= ?? 32 * 54 = ?? 20 * 92 = ?? 66 * 10 = ?? 5 * 19 = ?? 27 * 72 = ?? 87 * 66 = ?? 36 * 78
= ?? 62 * 15 = ?? 17 * 82 = ?? 67 * 95 = ?? 61 * 53 = ?? 56 * 92 = ?? 49 * 8 = ?? 36 * 1
0 = ?? 79 * 1 = ?? 59 * 64 = ?? 26 * 40 = ?? 42 * 62 = ?? 98 * 28 = ?? 11 * 58 = ?? 4 * 5
0 = ?? 94 * 52 = ?? 41 * 39 = ?? 32 * 46 = ?? 97 * 16 = ?? 88 * 65 = ?? 48 * 23 = ?? 5 *
37 = ?? 29 * 83 = ?? 15 * 68 = ?? 55 * 64 = ?? 73 * 28 = ?? 58 * 27 = ?? 71 * 93 = ?? 57
* 17 = ?? 18 * 81 = ?? 45 * 76 = ?? 64 * 10 = ?? 38 * 59 = ?? 58 * 60 = ?? 32 * 86 = ?? 4
3 * 34 = ?? 3 * 46 = ?? 41 * 82 = ?? 50 * 86 = ?? 67 * 24 = ?? 40 * 9 = ?? 23 * 100 = ??
50 * 81 = ?? 51 * 64 = ?? 82 * 100 = ?? 88 * 27 = ?? 3 * 90 = ?? 2 * 68 = ?? 6 * 84 = ??
9 * 5 = ?? 86 * 46 = ?? 9 * 11 = ?? 97 * 42 = ?? 62 * 93 = ?? 26 * 48 = ?? 94 * 69 = ?? 5
6 * 88 = ?? 22 * 47 = ?? 46 * 34 = ?? 6 * 14 = ?? 98 * 81 = ?? 31 * 67 = ?? 73 * 83 = ??
92 * 62 = ?? 44 * 79 = ?? 99 * 71 = ??
Congratulations, mission 33 has been successfully completed!
```

Analisi:

In questo livello il file nella libreria `Mathematics_101` conteneva le soluzioni del gioco, quindi si usa `<` che permette di passare un input personalizzato ad un comando che userà come `stdin`.

Quindi facendo `gsh check < path/Mathematics_101` si exploitava il gioco mandando automaticamente le risposte alle domande

Livello 34: Ricordiamo i libri di Merlin

Obiettivo della missione:

Mission goal

Merlin's old spell books are kept in his office, in the library. You need to save a list of all those spell books (and nothing else) in a file called "inventory.txt", in the drawer ...

Procedimento:

```
~/Castle/Main_building/Library/Merlin_s_office  
[mission 34] $ ls grimoire_* > Drawer/inventory.txt  
  
~/Castle/Main_building/Library/Merlin_s_office  
[mission 34] $ gsh check  
  
Congratulations, mission 34 has been successfully completed!
```

Analisi:

Per questo livello bastava listare tutti i grimori presenti nell'ufficio e indirizzare l'`stdout` in un file nel `Drawer` chiamato `inventory.txt`. Quindi:

```
ls grimoire_* > Drawer/inventory.txt
```

Livello 35: >, 2>, /dev/null/ e grep

Obiettivo della missione:

Mission goal

Merlin's old alchemy books are kept in his office, in the library. You need to output a list of all the books containing the alchemical compound 'gsh' (for "Glutathione").

Beware, it can be spelled with a mix of letters in upper and lowercase.

Some of those books are locked, and you are not allowed to consult them. Ignore them.

Remark

Your last command should display the list of books mentioning 'gsh', and nothing else. In particular, you should not show the chemical compounds themselves and no error message should appear.

Procedimento:

```
~/Castle/Main_building/Library/Merlin_s_office
[mission 35] $ grep 'gsh' -i -l grimoire_* 2>/dev/null
grimoire_ALQDwSHrPbqCNbJezm
grimoire_bkpGTWqADvAPugzqTsK
grimoire_CnfjsdnuynoUHVVGtakMy
grimoire_CnrkfoFYqEkSBG
grimoire_eWqXxnQaJPdBkWVTUYawmXiX
grimoire_gopojpIrLPwMZwJSpwImaLL
grimoire_HQDtaUyDgJDlxKCqmkPwZgfW0gu
grimoire_jIiHRrmNzgdMYJoAmbvyyZONA
grimoire_jJFrWCTxhsDqsKGzVuUkaYtKcvMDGU
grimoire_NYAnKNnZUALNOyBsWWxkrSiNBsE
grimoire_OakvbMKjRuAIyODmDaDZ
grimoire_oyNfcWwyqhEi
grimoire_qmdOizfdbLRqnaJNHcAnQBELdnUq
grimoire_SUDqJwKYJuRnEQRqSPdGf
grimoire_tBSGzDOfpMiP
grimoire_teUggAUmNdw
grimoire_UJvwqaQL
grimoire_UqyLYmqEFgFhrq
grimoire_VBcLoPtvozMuLQjIKkGuaVfIEoSqO
grimoire_VOFGCErWCFZwiCYqpXznkBCwCMX
grimoire_VovsIGIErJEVa
grimoire_WPDMYOWrVnpyUJby
grimoire_yFfVbebfisAjrD
grimoire_ymtHLMUdvpraAEvLOMqshna

~/Castle/Main_building/Library/Merlin_s_office
[mission 35] $ gsh check

Congratulations, mission 35 has been successfully completed!
```


Analisi:

COMMAND > FILE Invia i messaggi (**stdout**) al file. (sovrascrive il file)

COMMAND 2> FILE Invia i messaggi di errore (**stderr**) al file. (sovrascrive il file)

/dev/null File "virtuale" che è universalmente vuoto: qualsiasi cosa venga inviata lì sparisce.

grep Cerca righe, in uno o più file, che corrispondono al pattern specificato, l'output mostra le righe matchate. Flag importanti:

- **-i** : ignora maiuscole/minuscole
- **-v** : mostra le righe che non corrispondono al pattern
- **-r** : ricerca ricorsiva nelle directory
- **-l** : mostra solo i nomi file con almeno una corrispondenza

Per passare il livello dovevamo trovare i grimalti che contenevano "gsh" al loro interno, ignorandone il case. Quindi si usa **grep** per "gsh" con i flag **-i** e **-l**, che servono rispettivamente per renderlo case insensitive e filtrare l'output con solo i nomi dei file con almeno una corrispondenza, e gli diamo come file da considerare solo i grimalti quindi: **grep -i -l "gsh" grimoire_***. Ora il livello specificava che ci sarebbero stati file ai quali non si poteva accedere e quelli restituiscono un errore, quindi dobbiamo reindirizzare l'**stderr** da un'altra parte che non sia l'output del comando, e qua fa comodo **/dev/null** per cancellarlo praticamente:

grep -i -l "gsh" grimoire_* 2> /dev/null ed ecco fatto.

Livello 36: doppio forward, Merlin stava impazzendo

Obiettivo della missione:

```
Mission goal
=====

Merlin has turned crazy ... He paces around the observatory
tower and mumbles incoherently.

You need to filter out his ramblings to discover the secret
key he is the only one to know.

Hint
----

It is likely that Merlin's message is riddled with errors.
To complete this mission, you have to give the secret key
**using a file redirection**.
```

Procedimento:

```
~/Castle/Observatory
[mission 36] $ ./merlin > output.txt 2> errors.txt

~/Castle/Observatory
[mission 36] $ cat output.txt
THESECRETKEYISONSTDERR

~/Castle/Observatory
[mission 36] $ cat errors.txt
DtRkmDndBlxczgRFQbudzsXKNPwtEtEcDJzRGarQEVUSGhEHu1rKElkkFtZRu
huyfloyncPcBZpZcxdfaNegcNUtKECggECEgMqkPkyhScmtpEYGIwTVFLvUAZ

~/Castle/Observatory
[mission 36] $ gc < errors.txt
What is the secret key?
Congratulations, mission 36 has been successfully completed!
```

Analisi:

Questo livello sembrava un po' tricky, ma infondo `./merlin` diceva in `stdout` che la chiave fosse dentro `stderr` e in `stderr` c'era effettivamente la chiave.

Livello 37: `chmod` e i segreti del Re

Obiettivo della missione:

Mission goal

The door to the King's quarter is in the throne room.
Go to the King's quarter.

Remark

Access to this part of the castle is probably restricted.

Procedimento:

```
~  
[mission 37] $ cd Castle/Main_building/Throne_room/  
  
~/Castle/Main_building/Throne_room  
[mission 37] $ ls -l  
total 4  
drw-rw-r-- 2 kali kali 4096 Nov 15 13:31 Kings_quarter/  
  
~/Castle/Main_building/Throne_room  
[mission 37] $ chmod x Kings_quarter/  
chmod: invalid mode: 'x'  
Try 'chmod --help' for more information.  
  
~/Castle/Main_building/Throne_room  
[mission 37] $ chmod +x Kings_quarter/  
  
~/Castle/Main_building/Throne_room  
[mission 37] $ cd Kings_quarter/  
  
~/Castle/Main_building/Throne_room/Kings_quarter  
[mission 37] $ ls -A  
  
~/Castle/Main_building/Throne_room/Kings_quarter  
[mission 37] $ gsh check  
  
Congratulations, mission 37 has been successfully completed!
```

Analisi:

Qui viene introdotto il `chmod` che modifica i permessi di accesso su file e directory. Permessi principali: lettura (r), scrittura (w), esecuzione (x) per utente (u), gruppo (g), altri (o). nella sintassi si usa `+` o `-` davanti il permesso per concederlo o rimuoverlo. Flag utili:

- `-R` applica ricorsivamente alle sottodirectory
- `-v` stampa le modifiche effettuate

Nel livello ci bastava andare nella `Throne_room` e cambiare i permessi del `Kins_quarter` per poterci accedere, poi entrarci dentro.

Quindi `chmod +x Kings_quarter/` poi `cd Kings_quarter/`.

Livello 38: more `chmod`

Obiettivo della missione:

```
Mission goal
=====

The King is rumored to keep the combination to his safe written on a note in his room.
Find that combination.

Remark
=====

The King probably tried to make this note unreadable!
```

Procedimento:

```
~/Castle/Main_building/Throne_room/Kings_quarter
[mission 38] $ ls -A
note .secret_note

~/Castle/Main_building/Throne_room/Kings_quarter
[mission 38] $ ls -l
total 4
-rw-rw-r-- 1 kali kali 11 Nov 19 02:40 note

~/Castle/Main_building/Throne_room/Kings_quarter
[mission 38] $ ls -l -A
total 8
-rw-rw-r-- 1 kali kali 11 Nov 19 02:40 note
--w--w--- 1 kali kali 11 Nov 19 02:40 .secret_note

~/Castle/Main_building/Throne_room/Kings_quarter
[mission 38] $ chmod +r .secret_note

~/Castle/Main_building/Throne_room/Kings_quarter
[mission 38] $ cat .secret_note
1088553330

~/Castle/Main_building/Throne_room/Kings_quarter
[mission 38] $ gc
What's the combination to open the King's safe? 1088553330

Congratulations, mission 38 has been successfully completed!
```

Analisi:

Qui dovevamo cambiare il permesso di lettura ad un file invisibile e poi leggerne il contenuto: `chmod +r .secret_note` poi `cat .secret_note`

Livello 39: Spodestato il Re

Obiettivo della missione:

Mission goal

The safe in the throne room contains the king's crown.
Steal it and store it in your chest.

Also, the base of the crown is inscribed with a magical sequence of digits. Look at those digits and remember them.

Procedimento:

```
~/Forest/Hut/Chest
[mission 39] $ cd && cd Castle/Main_building/Throne_room && ls -l -A
total 8
drwxrwxr-x 2 kali kali 4096 Nov 19 02:40 Kings_quarter/
d----- 2 kali kali 4096 Nov 19 02:45 Safe/

~/Castle/Main_building/Throne_room
[mission 39] $ chmod +r+w+x Safe

~/Castle/Main_building/Throne_room
[mission 39] $ cd Safe

~/Castle/Main_building/Throne_room/Safe
[mission 39] $ ls -l -A
total 4
----- 1 kali kali 48 Nov 19 02:45 crown

~/Castle/Main_building/Throne_room/Safe
[mission 39] $ chmod +r+x+w crown && cat crown
  jgs
(^\\/^\\/^)
\@*\@*\@/
{_956_}

~/Castle/Main_building/Throne_room/Safe
[mission 39] $ mv crown ~/Forest/Hut/Chest && gc < 956
bash: 956: No such file or directory

~/Castle/Main_building/Throne_room/Safe
[mission 39] $ mv crown ~/Forest/Hut/Chest && gc
mv: cannot stat 'crown': No such file or directory

~/Castle/Main_building/Throne_room/Safe
[mission 39] $ gc
What are the 3 digits inscribed on the base of the crown? 956

Congratulations, mission 39 has been successfully completed!
```

Analisi:

Dentro la `Throne_room/` troviamo la `Safe/` che non ha permessi, quindi glie li aggiungiamo `chmod +r+w+x Safe` e ci entriamo dentro `cd Safe/` qui si trova la corona, sempre senza permessi quindi: `chmod +r+w+x crown && cat crown` facciamo anche stampare il contenuto per vederci il codice segreto da usare al momento del `gsh check`.

Livello 40: Un po' di `find` con un po' di `cat` con un po' di `mv`

Obiettivo della missione:

Mission goal

Look for the ruby in the maze in the garden, and move it to your chest.

Hint

The filename does not contain the string "ruby". Since there aren't many files in the maze, a possible strategy is to look for all `**file**` (as opposed to directories), and check by hand which one contains the string "ruby".

Procedimento:

```
~/Garden/Maze
[mission 40] $ find -type f
./aab166a4bf/22e710ac/4eaa733b5b52b5605a8620/36431
./aab166a4bf/a08808ef4f249a4c17693c62c3c8/7b7f1c6bdb9c0f64dbff/15756
./167337ddffe7055eb22d03571051/17f30e0658/315733b486893/17475
./8c9566353cc192567127/75e18986f2/ff1df473b53d9603251b908c/3426
./c492e17f878c66faecce1a6c/9d307db055a0d4e4c50df/f90fc4de0643/34826
./586e1f92629dab17c6/b5e62273d87b/e8ee76dfe3e431dab003441/50888

~/Garden/Maze
[mission 40] $ cat ./aab166a4bf/22e710ac/4eaa733b5b52b5605a8620/36431 ./aab166a4bf/a08808ef4f2
49a4c17693c62c3c8/7b7f1c6bdb9c0f64dbff/15756 ./167337ddffe7055eb22d03571051/17f30e0658/315733b4
86893/17475 ./8c9566353cc192567127/75e18986f2/ff1df473b53d9603251b908c/3426 ./c492e17f878c66fae
cce1a6c/9d307db055a0d4e4c50df/f90fc4de0643/34826 ./586e1f92629dab17c6/b5e62273d87b/e8ee76dfe3e4
31dab003441/50888
36431 stone 1063871f1a3bc1317f127e3299a146a27453ecd4
15756 stone eace371ca9ec3be36add722e4e14ce79d98548e4
17475 stone 2a3ee39f9f585f45d5430c775ece309c5d0ce9e3
3426 stone 2f2863d70d40f5e02225f1edaf864e7b8b04144f
34826 stone 23c49267200b76b4b5edf8686fdf59f8f3167f8c
50888 ruby 9ef6019b5b602f8992ac7b442e8739b8b33bafb5

~/Garden/Maze
[mission 40] $ mv ./586e1f92629dab17c6/b5e62273d87b/e8ee76dfe3e431dab003441/50888 ~/Forest/Hut/
Chest

~/Garden/Maze
[mission 40] $ gc

Congratulations, mission 40 has been successfully completed!
```

Analisi:

Qui il comando chiave è `find -type f` per matchare solo elementi di tipo file, poi diamo un'occhiata al contenuto dei file trovati, e cerchiamo quello che contiene ruby, poi ci si mette nella cesta e fine.

Livello 41: Benvenuto `xargs`

Obiettivo della missione:

```
Mission goal
-----

Combine several commands with "|" in order to find the diamond in
the maze, and move it to your chest.

Remark
-----

The filename doesn't contain the string "diamond". You need to
find a file that contains the string "diamond".
```

Procedimento:

```
~/Garden/Maze
[mission 41] $ find -type f | xargs grep 'diamond' -i -l
./233544795fd577b615d676318a69d7/b3535caf080d38554f/8bcc200b2a9a571cee05ea1b5e

~/Garden/Maze
[mission 41] $ find -type f | xargs grep 'diamond' -i -l | xargs mv ~/Forest/Hut/Chest
mv: cannot overwrite non-directory './233544795fd577b615d676318a69d7/b3535caf080d38554f/8bcc200
b2a9a571cee05ea1b5e' with directory '/home/kali/gameshell.2/World/Forest/Hut/Chest'

~/Garden/Maze
[mission 41] $ mv ./233544795fd577b615d676318a69d7/b3535caf080d38554f/8bcc200b2a9a571cee05ea1b5
e ~/Forest/Hut/Chest/

~/Garden/Maze
[mission 41] $ gc

Congratulations, mission 41 has been successfully completed!
```

Analisi:

xargs Costruisce ed esegue comandi prendendo argomenti dallo standard input. Utile per trasformare output in input di altri comandi, specie quando serve gestire molti argomenti e filtrarli.

In questo livello si combinano `find -type f | xargs grep 'diamond' -i -l` per controllare solo i file e trovare quello con la corrispondenza a “diamond”, poi ce lo mettiamo nella cesta. In questo caso l'utilizzo di **xargs** è fondamentale perché senza il **grep** agirebbe direttamente sulla risposta del **find** quindi cercando solo nel path o nome del file, mentre così cerca all'interno.

Livello 42: I debiti dell'ex Re

Obiettivo della missione:

Mission goal

Next to the castle, there is a merchant stall. People often buy on credit and reimburse their debt when they can. The shopkeeper keeps books on everyone's debt on a scroll. Whenever someone pays his debt, he inscribes "PAID" next to the corresponding transaction.

Combine several commands with ``|`` in order to find the King's debt.

Remark

You are only allowed 3 commands to find the King's debt. You can always reset the counter with `gsh reset`, but the whole stall and the debts of everyone will be re-generated as well.

Hint

When there are no sub-directories, an alternative to ``find . -name '*boring_object*'`` is to use `ls` and filter the result with grep:
\$ ls | grep "boring_object"

Procedimento:

```
(0)
~/Stall
[mission 42] $ ls | grep -v "boring_object"
d9a65fdaa0551_s_c_r_o_l_l_d9a65fdaa055135c
(1)
~/Stall
[mission 42] $ cat d9a65fdaa0551_s_c_r_o_l_l_d9a65fdaa055135c | grep " King " | grep -v "PAID"
the King bought a chackal for 5 coppers.
the King bought a bag flour for 6 coppers.
the King bought an apple for 3 coppers.
the King bought an opal for 3 coppers.
(2)
~/Stall
[mission 42] $ gc
How much does the king owe? 17

Congratulations, mission 42 has been successfully completed!

[ progress was saved in /home/kali/gameshell-save.sh ]
```


Analisi:

Abbiamo un limite di 3 comandi massimo.

Qui usiamo `ls | grep -v "boring_object"` per filtrare solo i file che non contengono "boring_object" nel nome, visualizziamo il contenuto con `cat` filtriamo le righe con la scritta " King " e poi quelle senza "PAID" e ci ritroviamo così i debiti del re:

```
cat scroll_name | grep " King " | grep -v "PAID"
```

Livello 43: Tutto su una riga

Obiettivo della missione:

Mission goal

Combine several commands with ``|`` in order to find the number of unpaid items.

Remark

You are only allowed a single command.

You can always reset the counter with `gsh reset`, but the whole stall will be re-generated.

Procedimento:

```
(0)
~/Stall
[mission 43] $ ls | grep -v "boring_object" | xargs grep -v "PAID" | wc -l | gc
How many unpaid items are there?
Congratulations, mission 43 has been successfully completed!
```

Analisi:

Questo livello vuole che la soluzione sia tutta su una singola riga.

Dobbiamo trovare il totale di oggetti che non sono ancora stati pagati quindi:

```
ls | grep -v "boring_object"
```

così troviamo lo scroll

```
ls | grep -v "boring_object" | xargs grep -v "PAID"
```

prendiamo il contenuto dello scroll e filtriamo solo gli oggetti non pagati

```
ls | grep -v "boring_object" | xargs grep -v "PAID" | wc -l | gc
```

con wc -l sull'output possiamo contare le righe totali, e l'output di questo comando lo diamo direttamente al gsh check

Livello 44: Caesar Shift Cipher

Obiettivo della missione:

```
Mission goal
=====

A secret message has been found, it is kept in the drawer in
Merlin's office. It was probably enciphered using a Caesar shift
cipher.

Decrypt it by making an exhaustive search from the command line.

Hint
----

All other secret messages that have been found were using a shift
between 10 and 16.
```

Procedimento:

```
~/Castle/Main_building/Library/Merlin_s_office/Drawer
[mission 44] $ cat secret_message | tr "a-z" "b-za-y" | tr "a-z" "b-za-y" | tr "a-z" "b-za-y" |
tr "a-z" "b-za-y" | tr "a-z" "b-za-y" | tr "a-z" "b-za-y" | tr "a-z" "b-za-y" | tr "a-z" "b-za-
-y" | tr "a-z" "b-za-y" | tr "a-z" "b-za-y" | tr "a-z" "b-za-y" | tr "a-z" "b-za-y"
here is my will:
you will get my chest, and everything it contains.
this chest is in the cellar, and the word to make
it re-appear is: dxzfz
merlin the enchanter

~/Castle/Main_building/Library/Merlin_s_office/Drawer
[mission 44] $ gc
What's the key that will make Merlin's chest to appear?
dxzfz

Congratulations, mission 44 has been successfully completed!
```

Analisi:

```
tr [opzioni] SET1 [SET2]
```

tr Traduce o elimina caratteri da un flusso di testo standard input e manda il risultato allo standard output. Utilizzato per sostituzioni semplici carattere per carattere o per manipolare flussi di testo. Flag importanti:

- **-d** elimina i caratteri in **SET1** invece di sostituirli
- **-s** comprime sequenze ripetute di caratteri in **SET1** in uno solo (es. più spazi diventano uno solo)
- **-c** complementa il set, cioè opera su tutti i caratteri non inclusi in **SET1**
- **-t** tronca **SET1** alla lunghezza di **SET2** (per metodi di sostituzione)

La key dentro il file **secret_message** è scriptata banalmente con uno shift di caratteri sull'ordine alfabetico di shift amount compreso tra 10 e 16.

Per applicare tale script sull'output UNA volta si scrive:

```
cat secret_message | tr "a-z" "b-za-y"
```

Quindi aggiungendo il pezzo **| tr "a-z" "b-za-y"** dalle 9 alle 15 volte. eseguendo, controlliamo l'output e se è sempre cifrato: ↑, ctrl+v, ↵.

Una volta che il messaggio è umanamente leggibile abbiamo trovato la chiave e finalmente finito il gameshell.sh

Livello 45: FINALMENTE!

