

C Piscine
Shell 00

Sommario: This document is the subject for the Shell 00 module of the C Piscine @ 42.

# Indice

•	1501 dzioiii	
II	Preambolo	3
III	Esercizio 00 : Z	5
IV	Esercizio 01 : testShell00	6
V	Esercizio 02 : Oh yeah, mooore	7
VI	Esercizio 03 : Connettimi!	8
VII	Esercizio 04 : midLS	9
VIII	Esercizio 05 : GiT commit	10
IX	Esercizio 06 : gitignore	11
$\mathbf{X}$	Esercizio 07 : diff	12
XI	Esercizio 08 : clean	13
XII	Esercizio 09 : Illusions, not tricks, Michael	14

#### Capitolo I

#### Istruzioni

- Fate riferimento solo a questa pagina: non fidatevi delle dicerie.
- Questo documento può subire variazioni prima della scadenza per la presentazione, verificate quindi il suo contenuto prima di presentare gli esercizi.
- Gli esercizi sono presentati seguendo un ordine di difficoltà crescente. Ai fini della valutazione NON si prendono in considerazione gli esercizi se i precedenti non sono stati completati correttamente.
- Controllate i permessi dei vostri file e delle vostre cartelle.
- Dovete seguire le procedure di presentazione per tutti gli esercizi.
- I vostri esercizi saranno controllati e valutati dai vostri compagni di corso.
- I vostri esercizi saranno soggetti, oltre alla valutazione tra pari, al controllo e alla valutazione da parte di un programma chiamato Moulinette.
- Moulinette sarà estremamente meticolosa e severa nel valutare il vostro lavoro. Essendo il suo un processo automatico senza possibilità di ricorso, assicuratevi di essere il più precisi possibile al fine di evitare brutte sorprese.
- Gli esercizi del modulo Shell devono essere eseguibili tramite /bin/sh.
- <u>NON</u> sarà tollerato <u>ALCUN</u> file aggiuntivo nelle cartelle presentate oltre a quelli specificati in questo documento.
- Dubbi o domande? Chiedi a chi si trova alla tua destra, altrimenti a chi si trova alla tua sinistra.
- I vostri manuali si chiamano Google / man / Internet / ....
- Date un occhiata alla sezione Piscina C del forum dell Intranet.
- Prestate attenzione agli esempi proposti, in quanto potrebbero mostrare dettagli non esplicitamente presentati nel documento...

#### Capitolo II

#### Preambolo

Here are the lyrics for City Hunter's theme song "Moonlight Shadow":

The last time ever she saw him
Carried away by a moonlight shadow
He passed on worried and warning
Carried away by a moonlight shadow.
Lost in a riddle that Saturday night
Far away on the other side.
He was caught in the middle of a desperate fight
And she couldn't find how to push through

The trees that whisper in the evening Carried away by a moonlight shadow Sing a song of sorrow and grieving Carried away by a moonlight shadow All she saw was a silhouette of a gun Far away on the other side.

He was shot six times by a man on the run And she couldn't find how to push through

[Chorus]

I stay, I pray
See you in Heaven far away...
I stay, I pray
See you in Heaven one day.

Four A.M. in the morning
Carried away by a moonlight shadow
I watched your vision forming
Carried away by a moonlight shadow
A star was glowing in the silvery night
Far away on the other side
Will you come to talk to me this night
But she couldn't find how to push through

[Chorus]

C Piscine

Shell 00

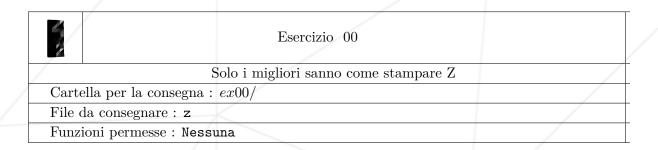
Far away on the other side.

Caught in the middle of a hundred and five
The night was heavy and the air was alive
But she couldn't find how to push through
Carried away by a moonlight shadow
Carried away by a moonlight shadow
Far away on the other side.

Unfortunately, this topic has nothing to do with City Hunter.

## Capitolo III

Esercizio 00: Z

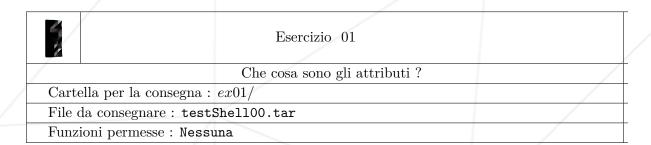


• Creare un file dal nome z che restituisca "Z", seguito da una nuova linea, ogni volta che viene usato come argomento del comando cat.

?>cat z
Z
?>

#### Capitolo IV

#### Esercizio 01: testShell00



- Creare un file dal nome testShell00 nella vostra cartella di consegna.
- Fare in modo di ottenere il seguente output(ad eccezione della linea "total 1"):

```
%> ls -l
total 1
-r--r-xr-x 1 XX XX 40 Jun 1 23:42 testShell00
%>
```

• Una volta riusciti nell'intento, lanciare eseguire il seguente comando per ottenere il file da consegnare: tar -cf testShell00.tar testShell00.



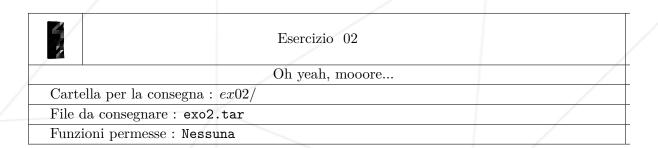
Non preoccupatevi di cosa ottenete al posto di "XX".



L'anno sarà accettato al posto dell'orario nel timestamp del file.

## Capitolo V

#### Esercizio 02: Oh yeah, mooore...



• Creare le seguenti cartelle e file. Fare in modo che l'output del comando 1s -1 eseguito nella cartella sia:

```
%> ls -l
total XX
drwx--xr-x 2 XX XX XX Jun 1 20:47 test0
-rwx--xr-- 1 XX XX 4 Jun 1 21:46 test1
dr-x---r-- 2 XX XX Jun 1 22:45 test2
-r----r-- 2 XX XX 1 Jun 1 23:44 test3
-rw-r---x 1 XX XX 2 Jun 1 23:43 test4
-r----r-- 2 XX XX 1 Jun 1 23:44 test5
lrwxr-xr-x 1 XX XX 5 Jun 1 22:20 test6 -> test0
%>
```

• Una volta riusciti nell'intento, lanciare eseguire il seguente comando per ottenere il file da consegnare: tar -cf exo2.tar \*.



Non preoccupatevi di cosa ottenete al posto di "XX".



L'anno sarà accettato al posto dell'orario nel timestamp del file.

## Capitolo VI

#### Esercizio 03: Connettimi!



- Assicuratevi di avere un ticket Kerberos valido(non scaduto). Se non ne avete uno ottenetelo.
- Una volta riusciti nell'intento, scrivere una lista di tutti i vostri ticket in un file dal nome klist.txt.



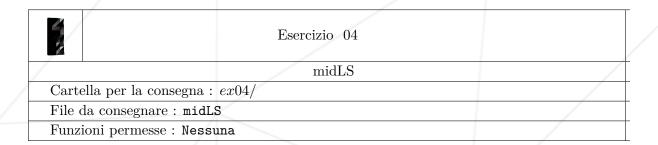
Il nome del file non é stato scelto a caso.



Questi comandi saranno utili più avanti, cercate di non dimenticarli!

#### Capitolo VII

Esercizio 04: midLS



• Mettete in un file dal nome midLS il comando di linea che stampa a video tutte le cartelle e i file(a parte quelle/i nascosti/e o ogni file/cartella che iniziano con un punto - si, anche quelli che iniziano con due punti ) presenti/e nella cartella attuale, separati da virgole e ordinati secondo data di creazione. Assicuratevi che i nomi delle cartelle siano preceduti da uno slash.



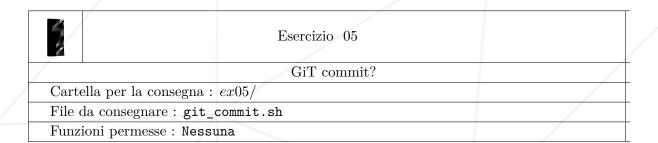
Tutto quello che non é stato richiesto non va fatto!



RTFM!(LIFM!)

## Capitolo VIII

Esercizio 05: GiT commit



• Creare uno script shell che stampi a video gli id degli ultimi 5 commit della vostra repository git.

```
%> bash git_commit.sh | cat -e
baa23b54f0adb7bf42623d6d0a6ed4587e11412a$
2f52d74b1387fa80eea844969e8dc5483b531ac1$
905f53d98656771334f53f59bb984fc29774701f$
5ddc8474f4f15b3fcb72d08fcb333e19c3a27078$
e94d0b448c03ec633f16d84d63beaef9ae7e7be8$
%>
```

Per testare il vostro script, utilizzeremo il nostro ambiente.



## Capitolo IX

Esercizio 06: gitignore



• In questo esercizio, scriverete un breve script shell che stampi a video tutti i file esistenti ignorati dalla vostra repository GiT. Esempio:

```
%> bash git_ignore.sh | cat -e
.DS_Store$
mywork.c~$
%>
```

Per testare il vostro script, utilizzeremo il nostro ambiente.



#### Capitolo X

Esercizio 07: diff



Esercizio 07

Cartella per la consegna : ex07/

File da consegnare : b

Funzioni permesse : Nessuna

• Creare un file dal nome b, in modo che :

%>diff a b > sw.diff



man patch

## Capitolo XI

Esercizio 08 : clean



#### Esercizio 08

Cartella per la consegna : ex08/File da consegnare : clean Funzioni permesse : Nessuna

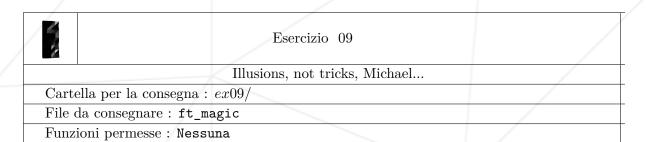
- $\bullet$  In un file dal nome clean mettete la linea di comando che effettua una ricerca per tutti i file presenti nella cartella attuale e tutte le sotto-cartelle il cui nome termina con  $\sim$  o inizia e finisce con #
- La linea di comando stamperà a video e cancellerà tutti i file trovati.
- Siete autorizzati ad utilizzare un unico comando: niente ';' o '&&' o altri raggiri.



man find

## Capitolo XII

Esercizio 09 : Illusions, not tricks, Michael...



• Creare un file magico dal nome ft\_magic formattato in modo da poter essere utilizzato per individuare file di tipo 42 file, che incorporano una stringa "42" al 42esimo byte.



man file