



CONCORSO PUBBLICO, PER ESAMI, A N. 1 POSTO DI CATEGORIA D - AREA TECNICA, TECNICO-SCIENTIFICA ED ELABORAZIONE DATI, CON RAPPORTO DI LAVORO SUBORDINATO A TEMPO INDETERMINATO PRESSO L'UNIVERSITÀ DEGLI STUDI DI MILANO - DIREZIONE SERVIZIO BIBLIOTECARIO D'ATENEO - CODICE 22451

La Commissione giudicatrice della selezione, nominata con Determina Direttoriale n. 8593 del 29/05/2024, composta da:

Dott. Franco Leveraro	Presidente
Dott.ssa Annalisa Corno	Componente
Dott. Matteo Savatteri	Componente
Dott.ssa Lucia Iacopino	Segretaria

comunica i quesiti relativi alla prova orale:

GRUPPO DI QUESITI n. 1

1 Il/La candidato/a illustri uno o più strumenti e/o procedure a propria scelta che possano essere utilizzati per garantire la sicurezza di dispositivi o reti di dispositivi informatici.

2 Il/La candidato/a commenti il seguente programma per computer, eventualmente concentrandosi su singole porzioni di codice.

```
#!/bin/bash

expr=".txt$"
count=0
count2=0

echo "relevant files..."

for i in $(ls)
do
    count=$((count + 1))
    echo $i | grep -q "$expr"
    [ $? -eq 0 ] && count2=$((count2 + 1))
Done

if [ $count -ne 0 ]
then
    echo -e "\nnumber of processed filenames: $count"
    echo "number of txt type files found: $count2"
else
    echo -e "\nno files found... for heaven's sake! Get to work!"
fi

exit 0.
```

3 Il/la candidato/a legga e traduca in italiano il seguente testo in lingua inglese.

Brano in inglese: This chapter frequently mentions disk devices, so we start with a brief overview of the concept of a device file.

A device special file corresponds to a device on the system. Within the kernel, each device type has a corresponding device driver, which handles all I/O requests for the device. A device driver is a unit of kernel code that implements a set of operations that (normally) correspond to input and output actions on an associated



piece of hardware. The API provided by device drivers is fixed, and includes operations corresponding to the system calls `open()`, `close()`, `read()`, `write()`, `mmap()`, and `ioctl()`.

The fact that each device driver provides a consistent interface, hiding the differences in operation of individual devices, allows for universality of I/O.

(Da "The Linux Programming Interface", pg 252).

GRUPPO DI QUESITI n. 2

1 Il/La candidato/a illustri una o più caratteristiche della famiglia di sistemi operativi Microsoft Windows e della famiglia di sistemi operativi GNU/Linux, ed evidenzi le differenze più significative riscontrabili tra le due categorie.

2 Il/La candidato/a commenti il seguente programma per computer, eventualmente concentrandosi su singole porzioni di codice.

```
#!/bin/bash

int_number=""

read -p "insert an integer number: " int_number

if [ "$int_number" = "" ]
then
    echo "$0 - no input :/ ... YOU are the ODD one..." >&2
    exit 1
fi

if [ $((int_number%2)) -eq 0 ]
then
    echo "$int_number is even"
else
    echo "$int_number is odd"
fi

echo "done"
exit 0
```

3 Il/la candidato/a legga e traduca in italiano il seguente testo in lingua inglese.

Brano in inglese: The Shadow Password File: `/etc/shadow`

Historically, UNIX systems maintained all user information, including the encrypted password, in `/etc/passwd`. This presented a security problem. Since various unprivileged system utilities needed to have read access to other information in the password file, it had to be made readable to all users. This opened the door for password-cracking programs, which try encrypting large lists of likely passwords (e.g., standard dictionary words or people's names) to see if they match the encrypted password of a user. The shadow password file, `/etc/shadow`, was devised as a method of preventing such attacks. The idea is that all of the nonsensitive user information resides in the publicly readable password file, while encrypted passwords are maintained in the shadow password file, which is readable only by privileged programs.

(Da "The Linux Programming Interface", pg 155).

GRUPPO DI QUESITI n. 3



1 Il/La candidato/a esponga le proprie conoscenze in merito al tema dei sistemi di virtualizzazione e containerizzazione, e se possibile fornisca alcuni esempi di implementazione.

2 Il/La candidato/a commenti il seguente programma per computer, eventualmente concentrandosi su singole porzioni di codice.

```
#!/bin/bash

countdown='10'

echo "Today is: - " `date`
echo -e "\nCountdown!\n"

while [ $countdown -ne 0 ]
do
    echo $countdown
    sleep 1
    countdown=$((countdown - 1))
done

case $(date +%d) in
17|24)
    echo "Not your day..."
    ;;
*)
    echo -e "\nBOOOOOM!\n"
    ;;
esac
exit 0
```

3 Il/la candidato/a legga e traduca in italiano il seguente testo in lingua inglese.

Brano in inglese: The system() function allows the calling program to execute an arbitrary shell command. In this section, we describe the operation of system(), and in the next section we show how system() can be implemented using fork(), exec(), wait(), and exit(). In Section 44.5, we look at the popen() and pclose() functions, which can also be used to execute a shell command, but allow the calling program to either read the output of the command or to send input to the command.

See main text for a description of return value

The system() function creates a child process that invokes a shell to execute command. Here is an example of a call to system():

```
system("ls | wc");
```

(Da "The Linux Programming Interface", pg 579).

Milano, 24 giugno 2024

La Commissione

Dott. Franco Leveraro - Presidente

Dott.ssa Annalisa Corno - Componente



UNIVERSITÀ DEGLI STUDI DI MILANO

Dott. Matteo Savatteri - Componente

Dott.ssa Lucia Iacopino - Segretaria