Esercitazione

1. Creare nella propria home una directory con nome testC

2. Controllare che permessi ha la directory appena creata

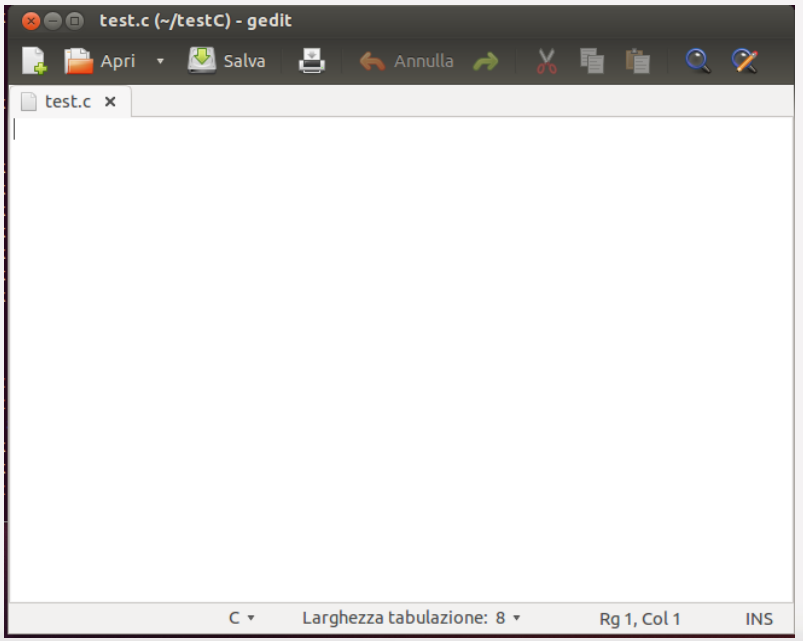
3. Andare all'interno della directory appena creata

4. Verificare i file presenti all'interno della directory appena creata.

Programmare in C su Linux

Dalla directory testC appena creata, eseguire il comando:

```
gedit test.c &
```



Simbolo per lanciare il programma in background.

Il primo programma C

Compilazione

Compilare il file test.c con il seguente comando:

```
$ gcc test.c -o test
```

Se non ci sono errori, il compilatore crea un file eseguibile test

Verificare che automaticamente questo file è stato creato con i permessi di esecuzione.

Esecuzione

Eseguire il programma test digitando:

$ ./test

```
giz@giz-HP-Pavilion:~/testC$ ./test
Inizio
/*FILE test.c*/
#include <stdio.h>

int main()
{
        printf("Inizio\n");
        system("cat test.c");
        printf("\nFine\n");
        return 0;
}

Fine
giz@giz-HP-Pavilion:~/testC$
```

Compilazione statica

Provare ora a compilare lo stesso sorgente con la seguente opzione:

```
$gcc -static test.c -o teststatic
```

Esecuzione teststatic

Eseguiamo ./teststatic che esegue esattamente quello che ha fatto il programmino test.

```
giz@giz-HP-Pavilion:~/testC$ ./teststatic
Inizio
/*FILE test.c*/
#include <stdio.h>

int main()
{
        printf("Inizio\n");
        system("cat test.c");
        printf("\nFine\n");
        return 0;
}

Fine
giz@giz-HP-Pavilion:~/testC$
```

Dimensione dei due eseguibili

Verificare all'interno della propria directory, la dimensione dell'eseguibile test e di teststatic:

```
-rwxrwxr-x 1 giz giz 755781 May 11 19:37 teststatic
-rwxrwxr-x 1 giz giz   7402 May 11 19:35 test
```

teststatic è molto più grande!

Librerie statiche e dinamiche

Nell'ambiente Linux le librerie si suddividono in due tipi:

- librerie statiche (*static libraries*) collegate al programma durante le fase di compilazione
- librerie dinamiche o condivise (*shared libraries*) che non sono contenute nel file eseguibile ma sono caricate e condivise mentre il programma è in esecuzione.

Librerie statiche vs dinamiche

Librerie Statiche	Librerie dinamiche
Le librerie diventano parte del programma per cui la sua dimensione cresce.	Le librerie dinamiche vengono collegate dinamicamente al programma e caricate solo quando servono. Il programma occupa meno spazio.
I programmi con librerie statiche si avviano velocemente, sono facili da distribuire e si evitano eventuali problemi di versione	Il programma impiega meno memoria. Se più programmi usano la stessa libreria, viene condivisa e non duplicata

Comando ldd

Il comando ldd stampa le dipendenze delle librerie condivise.

Eseguiamolo sui nostri due eseguibili:

```
giz@giz-HP-Pavilion:~/testC$ ldd teststatic
        not a dynamic executable
giz@giz-HP-Pavilion:~/testC$ ldd test
        linux-gate.so.1 =>  (0xb77bd000)
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb75fe000)
        /lib/ld-linux.so.2 (0xb77be000)
```

Roba da hacker!

Problem in Windows? Reboot.

Problem in Linux? Be root

L'utente root può tutto! Linux è stato creato per gli sviluppatori. Con debbuger, utility per fare 'trace' dei programmi, disassemblatori si può vedere ogni singola riga di ogni programma in esecuzione sul nostro computer ed è tutto già integrato nella distribuzione Linux!