

# LABORATORI DI INFORMATICA 2

ANNO ACCADEMICO 2002/2003

## MANUALE STUDENTE

### **Indice**

1. Gestione Sessione di lavoro
  - 1.1. Login al sistema
  - 1.2. Strumenti a disposizione dello studente
  - 1.3. Chiusura della sessione
2. Comandi Linux Principali
  - 2.1. Utilizzo del comando *man*
  - 2.2. Comandi per la gestione delle directory
  - 2.3. Comandi per la gestione dei file
  - 2.4. Invocazione programmi
3. Utilizzo della Console Didattica *i2*
  - 3.1. Avvio e Struttura della Console Didattica *i2*
  - 3.2. Login
  - 3.3. Assegnamento nuovo problema da risolvere
  - 3.4. Editazione e Compilazione della soluzione al problema assegnato
  - 3.5. Verifica della soluzione al problema assegnato
  - 3.6. Progressione dello stato dello studente
  - 3.7. Salvataggio dei file generati dallo studente
  - 3.8. Termine sessione di lavoro
  - 3.9. Altri comandi
4. Altro
  - 4.1. Supporti magnetici
  - 4.2. Riconoscimento diretto
5. Regole di Gestione dei Laboratori di Informatica 2
  - 5.1. regole generali
  - 5.2. regole per le sessioni di esercitazione
  - 5.3. regole per le sessioni di verifica

## Gestione Sessione di lavoro

### *Login al sistema*

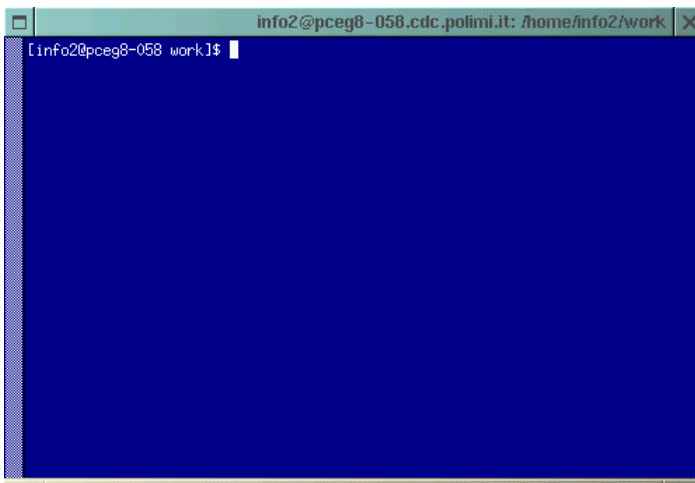
- Se il terminale richiede il Login al sistema Windows, premere il bottone **ShutDown** per riavviare il sistema (non spegnere le macchine!).
- Selezionare il sistema **Linux** tra le opzioni disponibili al riavvio del sistema.
- Il sistema **Linux** richiede il nome dell'utente e la password:
  - rispondere con **info2** alla richiesta del nome utente.
  - inserire **.info2** (punto + info2) alla richiesta della password (essa non viene visualizzata a video).
- Dopo l'autenticazione dell'utente, viene avviata una nuova sessione di lavoro direttamente in ambiente grafico.

### *Strumenti a disposizione dello studente*

- L'interfaccia grafica mette a disposizione, nella sua parte inferiore sinistra, un bottone per avviare un terminale grafico:

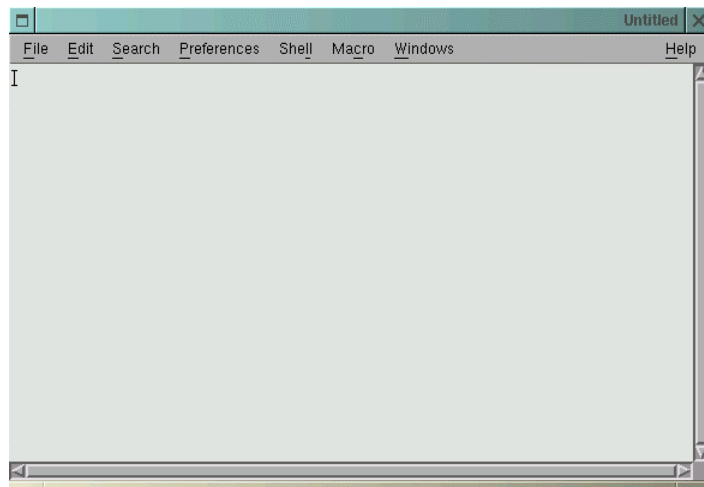


- apre, in una nuova finestra, un terminale dove invocare i comandi necessari allo svolgimento del laboratorio.



*Fig. 1: Finestra Terminale*

- Altri programmi necessari allo svolgimento degli esercizi di laboratorio sono:
  - **nedit**: per aprire un editor di testo da utilizzare per scrivere il codice richiesto.
  - **gcc**: per compilare i sorgenti scritti dallo studente.
  - **i2**: per avviare la Console Didattica.
- Tale comandi possono essere lanciati direttamente da una finestra terminale.
- Tutti i programmi eseguibili generati dallo studente devono essere lanciati da una finestra terminale.



*Fig. 2: Finestra Nedit*

### **Chiusura della sessione**

- Per lasciare l'ambiente grafico, posizionarsi sullo sfondo, premere il tasto destro del mouse e scegliere la voce **Termina Sessione**. Verrà riproposto il prompt di autenticazione dell'utente.
- Non spegnere il PC al termine della sessione di lavoro.

## Principali Comandi Linux

- I seguenti comandi sono invocabili dalla riga di comando di un qualsiasi terminale aperto sul desktop.

### Utilizzo del comando man

- **man** comando: visualizza la documentazione e la modalità di utilizzo del *comando* specificato.
- Il parametro *comando* può rappresentare anche una funzione di libreria del linguaggio *c*.
- E' consigliato abituarsi all'utilizzo del comando *man* per risolvere qualsiasi problema legato sia ai comandi del sistema Linux che alle funzioni del linguaggio *c*.

### Comandi per la gestione delle directory

- La directory predefinita per ogni sessione di laboratorio è **/home/info2**.
- Utilizzare **/home/info2/work** per memorizzare i file generati durante la sessione.
- Comandi utili:
  - **cd** *dir*: cambia la directory corrente.
  - **pwd**: restituisce la directory corrente.
  - **mkdir** *dir*: crea la directory *dir*.

### Comandi per la gestione dei file

- Comandi utili:
  - **ls -l**: elenca il contenuto della directory corrente.
  - **cp** *dir/file1 dir/file2*: copia *dir/file1* in *dir/file2*.
  - **mv** *dir/file1 dir/file2*: sposta o rinomina *dir/file1* in *dir/file2*.
  - **rm** *file*: rimuove *file*.

### Invocazione programmi

- Utilizzare **./fileexec** per lanciare un programma eseguibile *fileexec* non contenuto in directory predefinite, come i programmi implementati dallo studente presenti nella directory corrente.
- Utilizzando le frecce direzionali sul prompt dei comandi è possibile recuperare comandi lanciati precedentemente sullo stesso terminale.

## Utilizzo della Console Didattica i2

### Avvio e Struttura della Console Didattica i2

- Dal desktop, aprire un nuovo terminale tramite l'apposito bottone (doppio click) e sulla riga di comando digitare **i2**, viene avviata la Console Didattica:



Fig. 3: Console Didattica

- La finestra principale dell'interfaccia e' divisa in 5 aree:
  - La *Barra dei Menu* che contiene i menu *Console* e *Help*.
  - La *Barra dei Comandi* che raccoglie i comandi utente, descritti nelle sezioni successive.
  - La finestra *Messaggi del Verificatore*, sulla quale vengono mostrati i messaggi che il verificatore genera durante la fase di verifica di un problema. La finestra viene cancellata ad ogni nuova richiesta di verifica di un problema.
  - La finestra *Messaggi del Server Didattico*, in cui vengono visualizzati i messaggi che il server didattico invia alla console studente. Tali messaggi sono principalmente di servizio ed indicano eventuali errori nell'esecuzione dei comandi.
    - I messaggi di normale importanza sono mostrati in nero
    - I messaggi di particolare rilevanza sono visualizzati in rosso
    - I messaggi di assoluta rilevanza vengono mostrati in rosso e riportati anche in una finestra di popup.
  - La *Barra di Stato* che fornisce una indicazione sintetica sullo stato dell'interfaccia in ogni istante.

- Il menù Console contiene tre comandi di utilità generale. Tali comandi sono descritti nel seguito:
  - *Apri X Terminal*: apre un nuovo X Terminal cioè una finestra in cui e' in esecuzione una shell. In un X Terminal e' possibile utilizzare tutti i comandi UNIX, compresi i comandi di compilazione. Le soluzioni prodotte devono essere eseguite all'interno di un X Terminal.
  - *Apri Editor*: apre una nuova finestra dell'editor NEdit. Si noti che l'editor e' indipendente dall'interfaccia grafica.
  - *Esci*: abbandona l'interfaccia grafica. Questo comando ha il solo scopo di consentire il riavvio della consolle in caso di errore. La chiusura dell'interfaccia non comporta il logout dalla sessione. E' possibile, in un secondo momento, riaprire l'interfaccia grafica e continuare la sessione.
  
- Il menu Help raccoglie tutta la documentazione di cui gli studenti possono usufruire. I contenuti sono elencati di seguito:
  - *Consolle*: mostra la documentazione sulla consolle didattica stessa.
  - *Regolamento*: mostra il regolamento del laboratorio.
  - *Socket*: raccoglie le pagine del manuale UNIX sulle funzioni di gestione dei socket.
  - *Processi*: raccoglie le pagine del manuale UNIX sulle funzioni di gestione dei processi.
  - *Funzioni varie*: raccoglie alcune pagine del manuale UNIX su funzioni di utilità generale, quali funzioni di output e di formattazione di stringhe, funzioni di conversione e funzioni per la manipolazione delle stringhe.
  - *About*: mostra le informazioni sulla versione dell'interfaccia e della consolle i2.
  
- Il *Barra dei Comandi* riporta i comandi elencati di seguito:
  - *Login*
  - *Logout*
  - *Stato*
  - *Nuovo Problema*
  - *Mostra Problema*
  - *Allegati*
  - *Salva Sorgenti*
  - *Carica Sorgenti*
  - *Verifica*
  - *Termina Verifica*
  - *Test Server*

## Login

- Il comando *Login* inizia una sessione di lavoro. Il comando richiede il numero di matricola dello studente attraverso un'apposita finestra di dialogo.
- Perché lo studente si possa collegare alla Console Didattica, il responsabile deve avere aperto la sessione di laboratorio per la classe alla quale lo studente appartiene. Se la sessione di laboratorio globale non è aperta, non è possibile interagire con il server didattico.

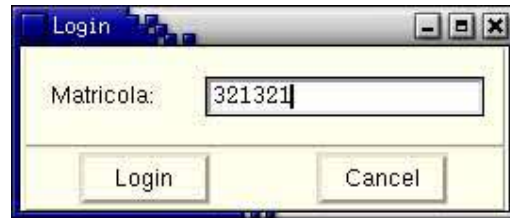


Fig. 4: Login

Inoltre:

- Lo studente, in casi eccezionali, può chiedere al responsabile di laboratorio di frequentare alcune sessioni di laboratorio in una classe diversa da quella di appartenenza.
- Non è permesso agli studenti di svolgere più sessioni di laboratorio rispetto a quelle effettuate dalla classe di riferimento. Non è possibile quindi anticipare sessioni di laboratorio rispetto al calendario predefinito.
- Esiste un numero massimo di sessioni di laboratorio effettuabili per ogni studente.
- L'operazione di Login può essere eseguita più volte in una stessa sessione se la sessione stessa non viene esplicitamente chiusa. E' necessario però che essa sia invocata sempre dalla stessa postazione di lavoro.
- Nel caso sia necessario cambiare la postazione utilizzata dallo studente in una sessione di laboratorio, prima di rieseguire il Login da un'altra postazione è necessario avvertire il responsabile di laboratorio per abilitare questo tipo di operazione. La nuova postazione di laboratorio utilizzata non deve essere attualmente in uso oppure non deve essere stata usata da un altro studente nella stessa sessione di laboratorio.

## Assegnamento nuovo problema da risolvere

- Il comando *Nuovo Problema* richiede al server didattico il testo di un nuovo problema da risolvere. Il testo del nuovo problema è mostrato in una finestra dedicata.

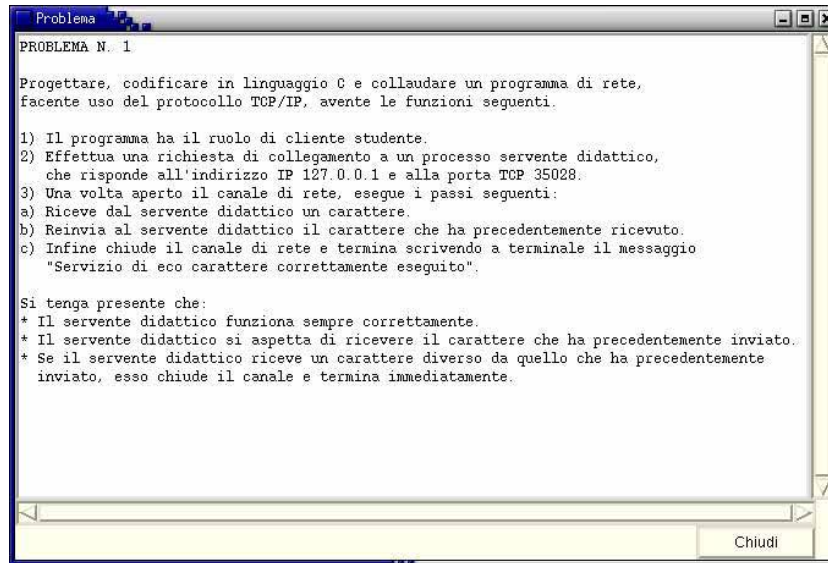


Fig. 5: Visualizzazione del testo di un problema assegnato

- Dopo aver chiuso la finestra con il testo del problema è possibile riaprirla con il comando *Mostra Problema* (anche in sessioni successive).
- Alcuni problemi richiedono dei file allegati (soluzioni parziali, file di supporto, eccetera) che saranno automaticamente scaricati nella directory di lavoro della postazione dello studente.
- In caso di cancellazione accidentale degli allegati è possibile recuperarli utilizzando il comando *Allegati*.
- Un problema assegnato deve essere necessariamente risolto prima di passare al successivo (in questo caso, lo stato dei pulsanti della Console Didattica non permette di richiedere un nuovo problema).

Inoltre:

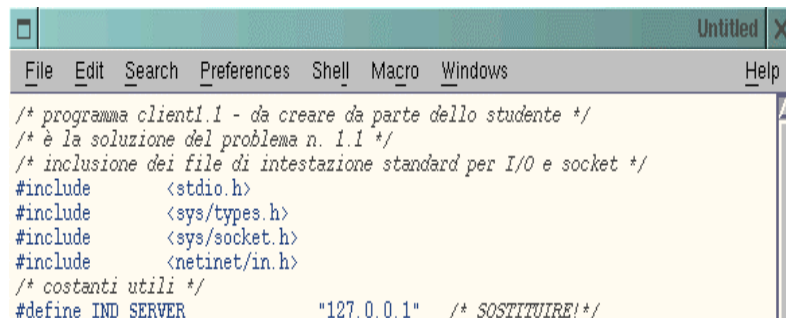
- Ad ogni sessione di laboratorio viene associato un livello massimo raggiungibile da ogni studente. Quindi è possibile che il sistema non assegni più problemi allo studente se questo ha raggiunto il livello massimo conseguibile nella sessione di laboratorio attiva.
- E' invece possibile che il sistema proponga agli studenti che hanno raggiunto il livello massimo conseguibile nella sessione di laboratorio attiva una serie di problemi facoltativi. Questi problemi non hanno effetto sulla progressione del livello degli studenti e non devono essere svolti per forza. Eventualmente uno studente in questa condizione può decidere di saltare una o più sessioni di laboratorio se decidesse di non esercitarsi con tali problemi e di ripresentarsi in laboratorio alla prima sessione che gli permetterà di avanzare di livello.
- Come conseguenza di un eventuale cambio della postazione utilizzata dallo studente, è possibile che alcuni parametri del problema assegnato allo studente siano modificati dal sistema.



## Edizione e Compilazione della soluzione al problema assegnato

- Utilizzare l'editor **nedit** per scrivere il codice richiesto. Esso mette a disposizione una serie di menù con i classici comandi di gestione del testo:
  - Menù File: open, close, save, new, exit.
  - Menù Edit: cut, copy, paste, delete.
  - Menù Search: find, replace, goto line number.
  - Menù Preferences: show line number (visualizza il numero delle linee di fianco al testo editato).
  - Menù Windows: lista dei file aperti.

Fig 6: editazione della soluzione del problema assegnato



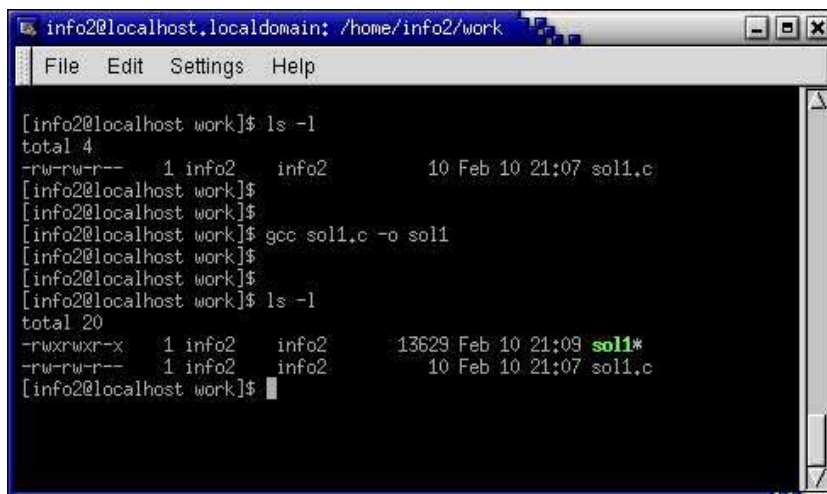
```

/* programma clienti.1 - da creare da parte dello studente */
/* è la soluzione del problema n. 1.1 */
/* inclusione dei file di intestazione standard per I/O e socket */
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
/* costanti utili */
#define IND SERVER "127.0.0.1" /* SOSTITUIRE!*/
  
```

- Una volta completata la scrittura del codice relativo alla soluzione del problema assegnato, salvare il file su disco, assegnandogli un nome sensato e chiaro.
- Aprire un nuovo terminale tramite l'apposito bottone sul desktop (doppio click) e spostarsi nella directory di lavoro desiderata.
- In questa nuova finestra, sulla riga di comando, utilizzare il compilatore **gcc** con la seguente sintassi:

`gcc file.c -o fileexec`

dove *file.c* è il file contenente il codice sorgente del programma, appena scritto, mentre *fileexec* è il nome del file eseguibile generato dalla fase di compilazione.



```

info2@localhost.localdomain: /home/info2/work
File Edit Settings Help

[info2@localhost work]$ ls -l
total 4
-rw-rw-r-- 1 info2 info2 10 Feb 10 21:07 sol1.c
[info2@localhost work]$
[info2@localhost work]$ gcc sol1.c -o sol1
[info2@localhost work]$
[info2@localhost work]$ ls -l
total 20
-rwxrwxr-x 1 info2 info2 13629 Feb 10 21:09 sol1*
-rw-rw-r-- 1 info2 info2 10 Feb 10 21:07 sol1.c
[info2@localhost work]$
  
```

Fig 7: Compilazione

- Il compilatore, se necessario, genera una lista di errori. Per ciascuno di essi viene riportato il numero di riga nel *file.c* e la descrizione dell'errore.

## Verifica della soluzione al problema assegnato

- Il comando *Verifica* inizia il processo di verifica di una soluzione sviluppata dallo studente da parte del server di verifica. Tale operazione disabilita tutti gli altri comandi, tranne quello di terminazione forzata della verifica, fino al completamento della verifica stessa.
- I messaggi inviati dal verificatore sono mostrati nella finestra superiore dell'interfaccia, mentre l'esito della verifica è mostrato nella finestra inferiore e in una finestra di dialogo. Tali messaggi possono riguardare:
  - richieste di attivazione dei programmi da verificare.
  - operazioni di verifica eseguite.
  - il risultato della verifica.
- Per lanciare il programma da verificare, implementato dallo studente, attivare un terminale e posizionarsi nella directory di lavoro. Quindi digitare `./fileexec`, dove `fileexec` è il nome dell'eseguibile sviluppato e compilato dallo studente.

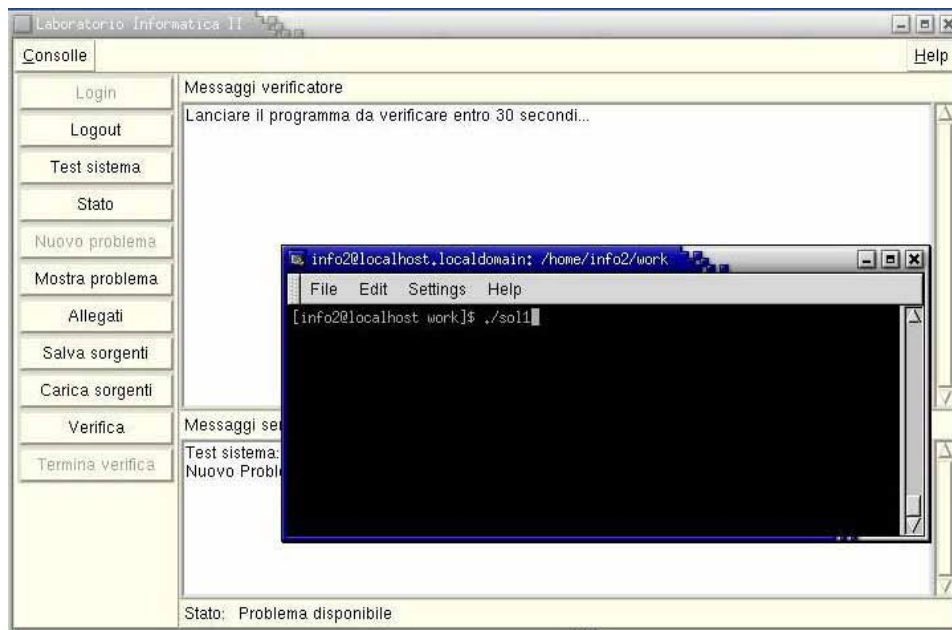


Fig. 8: Avvio della verifica del problema assegnato

Inoltre:

- Il processo di verifica di un problema può risultare lento a causa dei timeout standard impostati sulle varie operazioni da effettuare per verificare la correttezza della soluzione proposta dallo studente.
- *Termina verifica*: termina forzatamente il processo di verifica. A volte può accadere che la soluzione sviluppata possa causare un malfunzionamento del server di verifica. In tal caso, se dopo un certo periodo di tempo, il server non invia più alcun messaggio, può essere opportuno forzare la terminazione della verifica.
- Alcuni problemi facoltativi potrebbero non avere bisogno di un verificatore didattico. In questo caso è comunque necessario selezionare il comando *Verifica* prima di passare al problema successivo.

## Progressione dello stato dello studente

- Il comando *Stato* mostra lo stato consolidato dello studente.



Fig 9: Stato dello studente

## Salvataggio dei file generati dallo studente

- Il comando *Salva Sorgenti* permette di salvare i file sorgente sviluppati da ciascuno studente in una sessione di laboratorio. I file che possono essere salvati devono avere estensione *.c* e non devono superare complessivamente la dimensione di 16 KB. Per selezionare i file da salvare il comando *Salva Sorgenti* mostra una finestra di dialogo in cui è possibile gestire una lista di file.

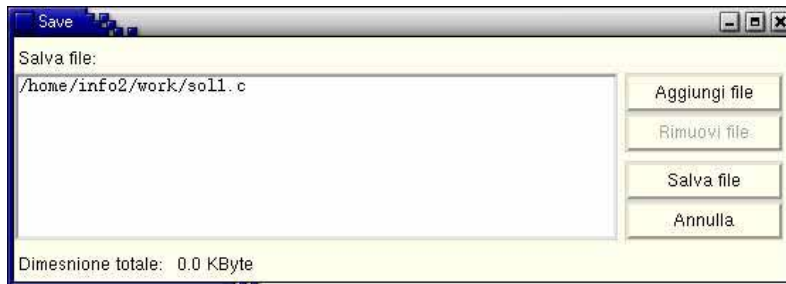


Fig 10: Elenco dei file da salvare

- Il pulsante *Aggiungi File* permette all'utente di selezionare dal file system locale i file da salvare sul server:

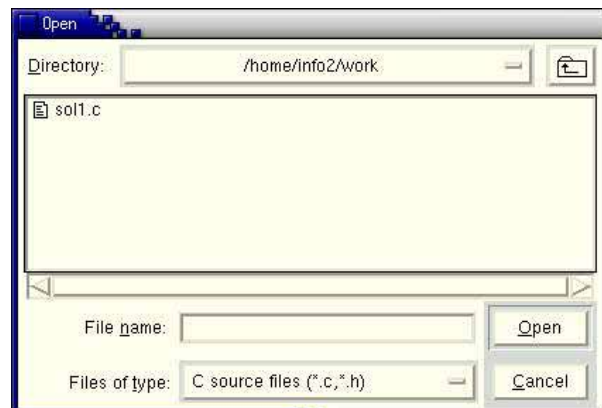


Fig. 11: Selezione dei file da salvare

- Si noti che l'operazione di salvataggio **sovrascrive in modo automatico e senza preavviso qualsiasi insieme di file precedentemente salvato**. In altre parole, si può salvare una sola versione dei propri file alla volta.
- Viceversa, il comando *Carica Sorgenti* permette di recuperare i file sorgenti salvati con l'ultima invocazione del comando *Salva Sorgenti*, anche se eseguito in una sessione precedente. Questi verranno scaricati nella directory corrente della postazione attuale dello studente.
- **I file recuperabili sono SOLO quelli salvati con l'ultima invocazione del comando *Salva Sorgenti***

### **Termine sessione di lavoro**

- La sessione di lavoro di tutti gli studenti che partecipano a una sessione di laboratorio di una certa classe può essere chiusa direttamente dal responsabile di laboratorio. Dopo tale chiusura lo studente non è più abilitato ad interagire con il server didattico.
- Prima della chiusura della sessione di laboratorio (circa 3 minuti prima) viene inviato un messaggio a tutti gli studenti coinvolti nella sessione per invitare gli stessi a salvare i propri programmi sul server didattico, tramite l'apposito comando della console didattica *Salva Sorgenti*.  
In questo intervallo di tempo, nessun altro comando della Console Didattica risulta abilitato.

Inoltre:

- Il comando *Logout* conclude in modo irreversibile una sessione di lavoro di uno studente, in qualsiasi momento della sessione stessa. Dopo la chiusura non è più possibile riaprire la sessione e continuare a lavorare.  
In caso di una chiusura erranea contattare il responsabile di laboratorio.
- Il comando *Esci* chiude semplicemente l'interfaccia grafica ma lascia inalterato lo stato di autenticazione dello studente. Sarà quindi possibile riconnettersi al server didattico riaprendo la Console Didattica e rieseguendo il comando Login.

### **Altri comandi**

- Il comando *Test Server* controlla se il server didattico è attivo. Nel caso il server sia terminato a causa di un possibile malfunzionamento è necessario effettuare un nuovo login, chiudendo e riavviando la Console Didattica.

## **Altro**

### **Supporti magnetici**

- Non è possibile copiare i file su supporti di memorizzazione removibili.

### **Riconoscimento diretto**

- Durante la sessione di laboratorio, è necessario esibire il tesserino di riconoscimento.

## Regole di Gestione dei Laboratori di Informatica 2

### **Regole generali:**

- Le 6 sessioni globali di laboratorio sono divise in due cicli: le prime 4 sessioni sono di tipo “esercitazione”, le ultime 2 di tipo “verifica”.

### **Regole per sessioni di esercitazione:**

- Nelle prime 4 sessioni si possono portare appunti, libri, ecc...;
- Nelle sessioni di esercitazione viene proposta una sequenza di problemi, uguali per tutti gli studenti.
- Il superamento di un certo numero predefinito di problemi vale 3 punti ed è obbligatorio (e non recuperabile); il livello dello studente diventa 1.
- Lo studente che finisce i problemi obbligatori prima della fine della quarta sessione può continuare ad esercitarsi con dei problemi facoltativi, ma non può anticipare problemi di tipo “verifica”; il suo livello rimane 1.
- Lo studente che finisce i problemi obbligatori entro la terza sessione e non intende avvalersi degli ulteriori esercizi facoltativi di verifica non è tenuto a frequentare le sessioni di esercitazione mancanti.
- Lo studente che non finisce i problemi obbligatori entro la quarta sessione (cioè rimane a livello 0) deve completare tali problemi durante le sessioni di verifica.

### **Regole per sessioni di verifica:**

- Nelle ultime 2 sessioni non si può portare niente e non si può esportare niente; viene fornita documentazione in linea e viene esercitato un controllo tipo esame
- Nelle sessioni di verifica lo studente deve risolvere 3 problemi, passando ai livelli 2, 3 e 4.
- Viene considerata la **prima** sessione di verifica di uno studente quella nella quale viene richiesto il primo problema di tipo *verifica*.  
Se questo assegnamento avviene, per esempio, solo a cinque minuti dal termine della sessione di laboratorio, l'intera prima sessione di verifica viene considerata come usufruita. Questo caso si verifica quando lo studente deve ancora risolvere uno o più problemi di esercitazione, nonostante sia già stata aperta una sessione di laboratorio di tipo *verifica*.