

Prontuario per Xcode



Guida rapida a Xcode in C di Marco Cartellà

In collaborazione con:

prof. Elio Piccolo
prof. Antonio Servetti.

Contatto: marco.cartella@studenti.polito.it



Indice

1	Introduzione	3
1.1	Contesto di utilizzo	3
1.2	Che cosa è Xcode?	3
2	Per iniziare	4
2.1	Download e installazione	4
3	Guida all'utilizzo	5
3.1	Creare un nuovo progetto in C	5
3.2	Aprire un progetto già creato	7
3.3	Interfaccia	8
3.4	Debug	11
3.5	Linea di comando	12
3.6	Eseguibile e linea di comando	13
3.7	Lettura e scrittura file	16
3.7.1	Creazione di un file .txt	16
3.7.2	Direttorio di lavoro	17
4	Dopo il corso	19
4.1	Disinstallazione	19
4.1.1	Procedura consigliata	19
4.1.2	Procedura alternativa	19
4.2	Rimozione parti poco utili di Xcode	19
5	Frequently Asked Questions - FAQ	21
5.1	Problemi con il debug	21
5.1.1	Vista assembler	21
5.2	Funzione System()	22

Scritto con L^AT_EX

1 Introduzione

1.1 Contesto di utilizzo

Questo piccolo manuale nasce dalla necessità di coprire gli studenti del corso di Informatica, tenuto al Politecnico di Torino, che non possono utilizzare il software consigliato, per esercitarsi alla programmazione in C, poiché in possesso di un MAC. Infatti il software attualmente consigliato, Code::Blocks, tuttora non è totalmente compatibile con la nuova piattaforma Apple. In attesa di una soluzione non c'è da disperare, infatti ci viene in contro Xcode.

1.2 Che cosa è Xcode?

Apple fornisce agli utenti che utilizzano il sistema operativo MAC OS X un ambiente di sviluppo integrato (Integrated development environment, IDE) contenente una suite di tool per sviluppo software, Xcode, un software completo che permette lo sviluppo di programmi con diversi linguaggi, anche differenti dal C, come C++, Objective-C, Objective-C++, Java, AppleScript, Python, Ruby, Rex e Swift, con una varietà di modelli di programmazione inclusi e non limitati a Cocoa, Carbon e Java. Inoltre, tool di terze parti permettono il supporto a GNU Pascal, Free Pascal, Ada, C# e Perl. In Figura 1 è riportata l'icona di Xcode.



Figura 1 - Icona del software Xcode

Precedentemente era fornito gratuitamente in bundle con il sistema operativo a partire da Mac OS X 10.3 Panther, sebbene sia in grado di generare programmi per qualsiasi versione di Mac OS X. Di recente invece non è più in bundle con il sistema operativo ma è possibile scaricarlo gratuitamente dal Mac Apple Store.

In Xcode risiedono innumerevoli vantaggi che lo rendono uno strumento professionale e uno dei software di programmazione tecnologicamente più avanzati. Infatti, due caratteristiche di rilievo di questa suite sono:

- Supporto alla distribuzione in rete del lavoro di compilazione: usando Bonjour e Xgrid è in grado di compilare un progetto su più computer riducendo i tempi di compilazione.
- Supporta la compilazione incrementale: è in grado di compilare il codice mentre viene scritto. Oltre al vantaggio di ridurre il tempo di compilazione, questa caratteristica dà anche la possibilità di avvisare l'utente programmatore di un eventuale errore di sintassi in tempo reale e dare, inoltre, quando possibile, un suggerimento per la correzione dell'eventuale errore

2 Per iniziare

2.1 Download e installazione

Il software è scaricabile direttamente dal Mac App Store seguendo il seguente link (*clicare sul link*):
<https://itunes.apple.com/it/app/xcode/id497799835?mt=12>

oppure eseguendo una semplice ricerca nell'applicazione App Store inserendo nel campo di ricerca la stringa "Xcode".

Una volta *clccato su "Ottieni"* (bottone grigio) e successivamente *su "Installa l'app"* (bottone verde) richiederà di inserire le credenziali dell'account Apple, *eseguire l'accesso* e attendere che l'installazione sia conclusa. Sarà conclusa quando la barra progresso nel launchpad sarà completa: la dimensione del file è dell'ordine dei 2.5GB quindi ci impiegherà parecchio (in Figura 2). Mettetevi comodi e attendete...



Figura 2 - Launchpad durante il download di Xcode

Una volta scaricato, l'applicazione verrà installata automaticamente. Al termine dell'installazione l'applicazione "luccicherà" (vedi Figura 3).

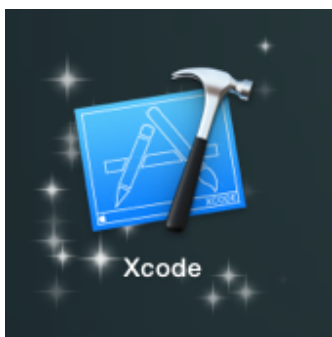


Figura 3 - Icona dopo l'installazione

3 Guida all'utilizzo

Di seguito, saranno riportate le principali caratteristiche del software utilizzate durante il corso. Si procederà, in primo luogo, introducendo l'interfaccia dell'ambiente di lavoro di Xcode, successivamente si mostrerà sommariamente, con alcuni screenshot passo-passo, le funzionalità necessarie per lo sviluppo dei codici:

- Creazione di un progetto C;
- Apertura di un progetto già creato;
- Debugging del codice;
- Passaggio dei parametri da linea di comando;
- Eseguire il codice da linea di comando;
- Lettura e scrittura file.

3.1 Creare un nuovo progetto in C

Per creare un nuovo progetto su Xcode basta *aprire Xcode*, una volta aperto apparirà la finestra iniziale (vedi Figura 4) dove vi è sulla sinistra 3 azioni da compiere e sulla destra la cronologia di file e progetti su cui si è lavorato in precedenza.

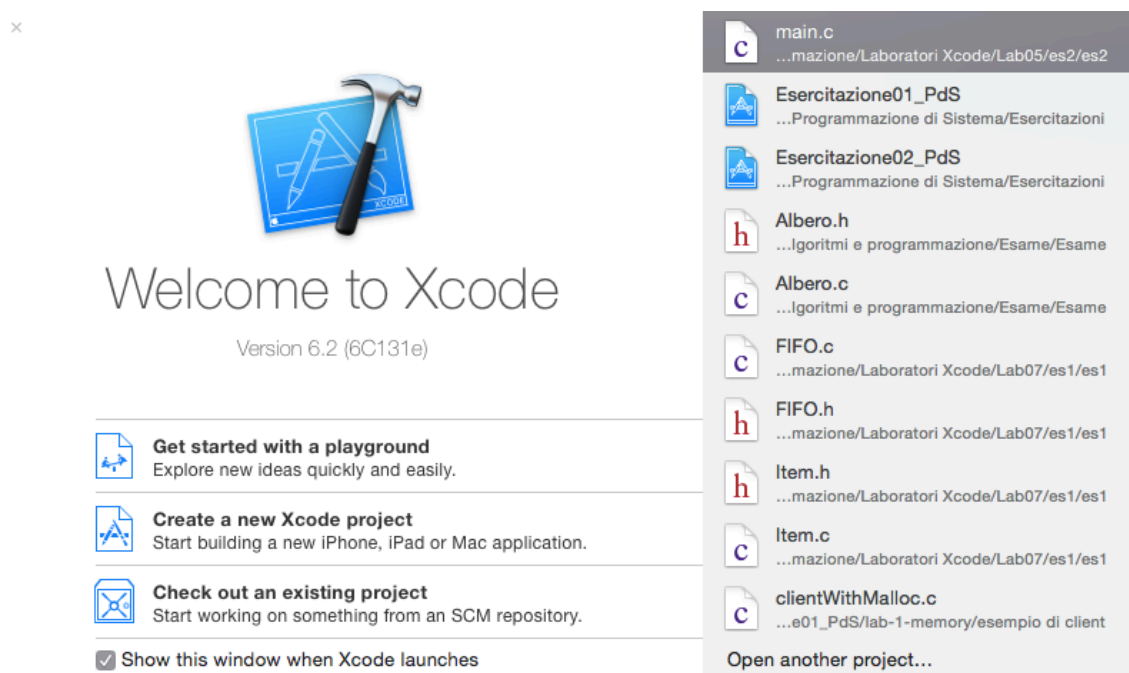


Figura 4 - Finestra di avvio

Una volta aperta la finestra clicare su "Create a new Xcode project" (vedi Figura 5).



Figura 5 - Creare un nuovo progetto

Si aprirà successivamente una finestra che vi chiederà il tipo di progetto che si vuole creare, selezionare "Command Line Tool" che si trova nel tab "Application" sotto "OS X", come in Figura 6.

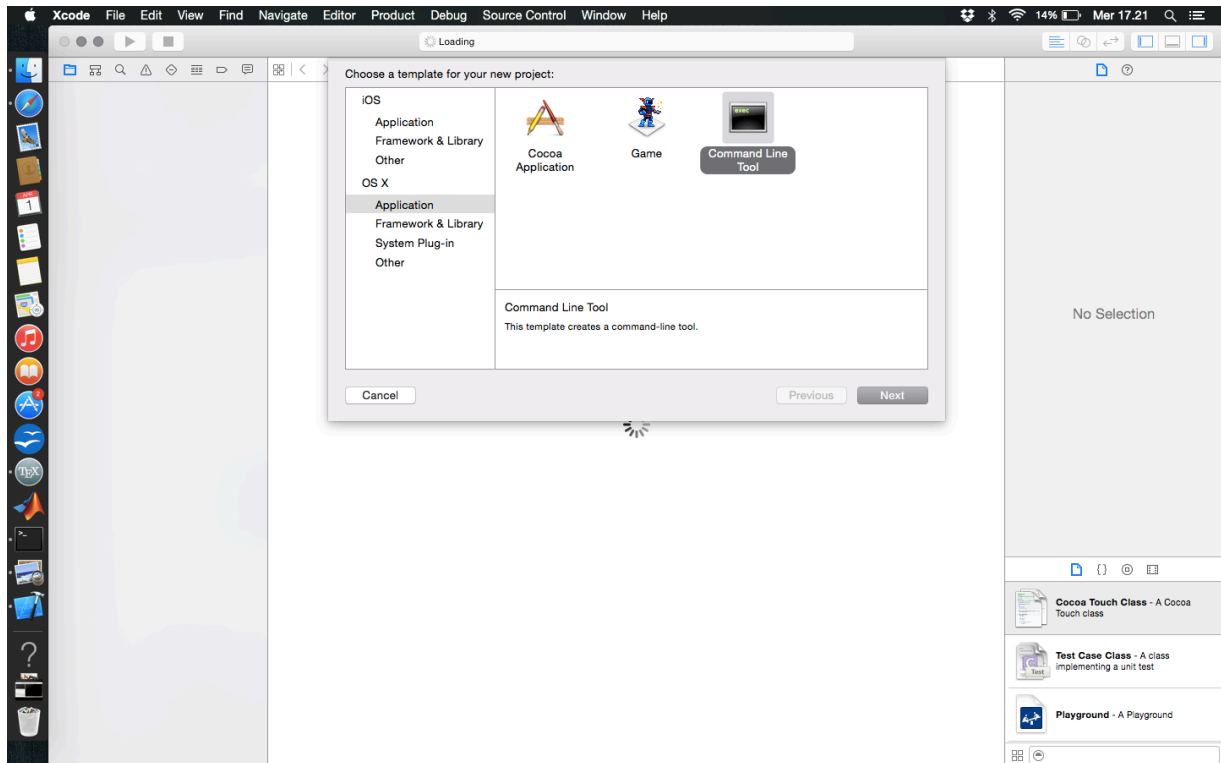


Figura 6 - Selezione tipo di Progetto

Cliccare su "Next" e riempire i seguenti campi come segue:

- Product Name: <nome progetto>
- Organization Name: <lasciare invariato>
- Organization Identifier: <lasciare invariato>
- Language: C

In Figura 7 è riportato un esempio.

Choose options for your new project:

Product Name:

Organization Name:

Organization Identifier:

Bundle Identifier:

Language:

Cancel Previous Next

Figura 7 - Nominare il progetto e selezione linguaggio

Cliccare nuovamente su *"Next"*.

IMPORTANTE: ricordarsi di selezionare il linguaggio C.

Successivamente, *selezionare la directory* nella quale desiderate inserire il progetto, muoversi nella posizione desiderata e cliccare sul bottone "Create". In Figura 8 è riportata la finestra per la selezione della directory.

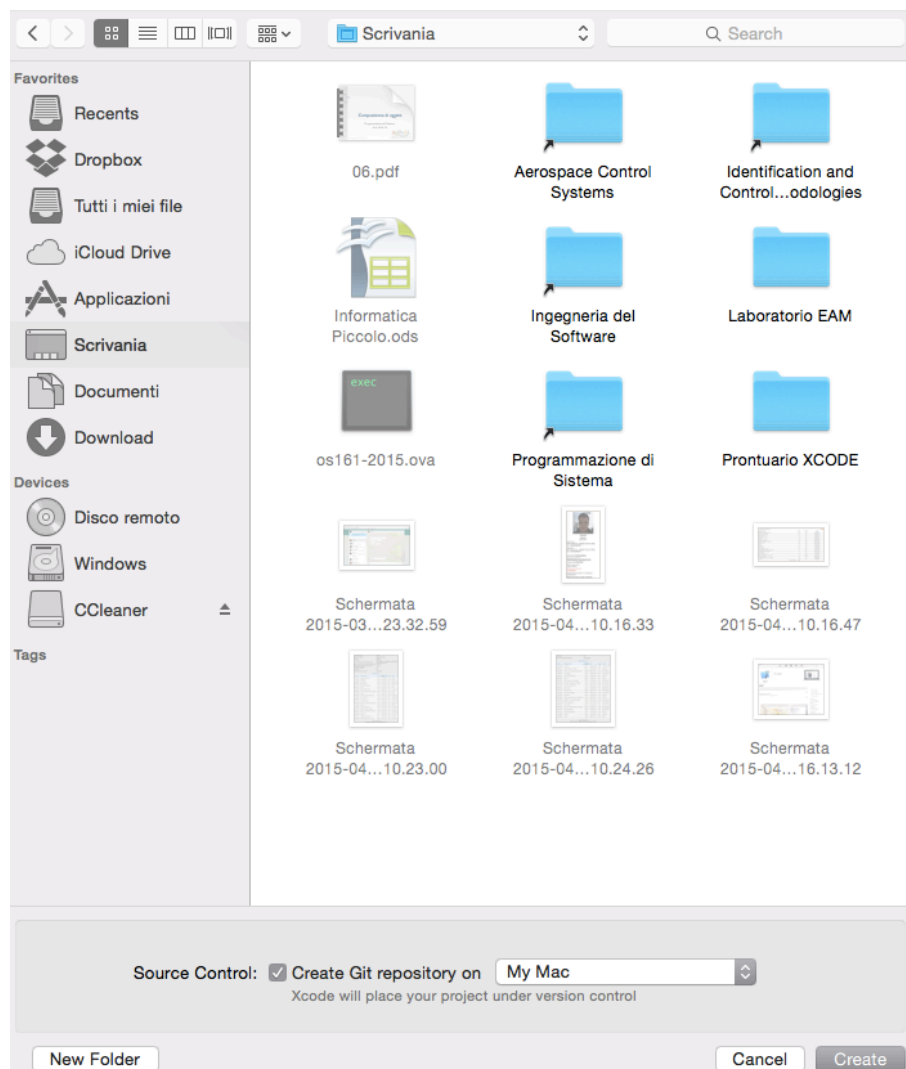


Figura 8 - Selezione directory

A questo punto, il progetto è creato. Si rimanda al paragrafo 3.3 per una descrizione essenziale dell'interfaccia.

3.2 Aprire un progetto già creato

Un primo errore commesso da programmatori neofiti è quello di aprire il file con estensione ".c" credendo basti per aprire e continuare lo sviluppo di una applicazione; in realtà ci sono molti altri file necessari allo sviluppo del codice. Quando si crea un progetto viene creata, solitamente, una cartella che contiene tutti questi file, uno di questi è un file che contiene l'intero progetto, su Xcode, ha estensione ".xcodeproj". Quindi come si può immaginare ogni editor (ad esempio Code::Blocks, Eclipse oppure Xcode) ha una sua gestione dei progetti e quindi non è possibile aprirli con un programma differente da quello utilizzato durante la creazione.

L'apertura di un progetto già creato la si svolge in due modi differenti, che si distinguono dal fatto di avere creato o meno il progetto con il proprio MAC o con uno differente:

- Metodo sempre utilizzabile:

Andare nella cartella contenente l'intero progetto e cliccare due volte sul file con estensione ".xcodeproj".

In Figura 9 è mostrato un esempio di contenuto della cartella progetto.

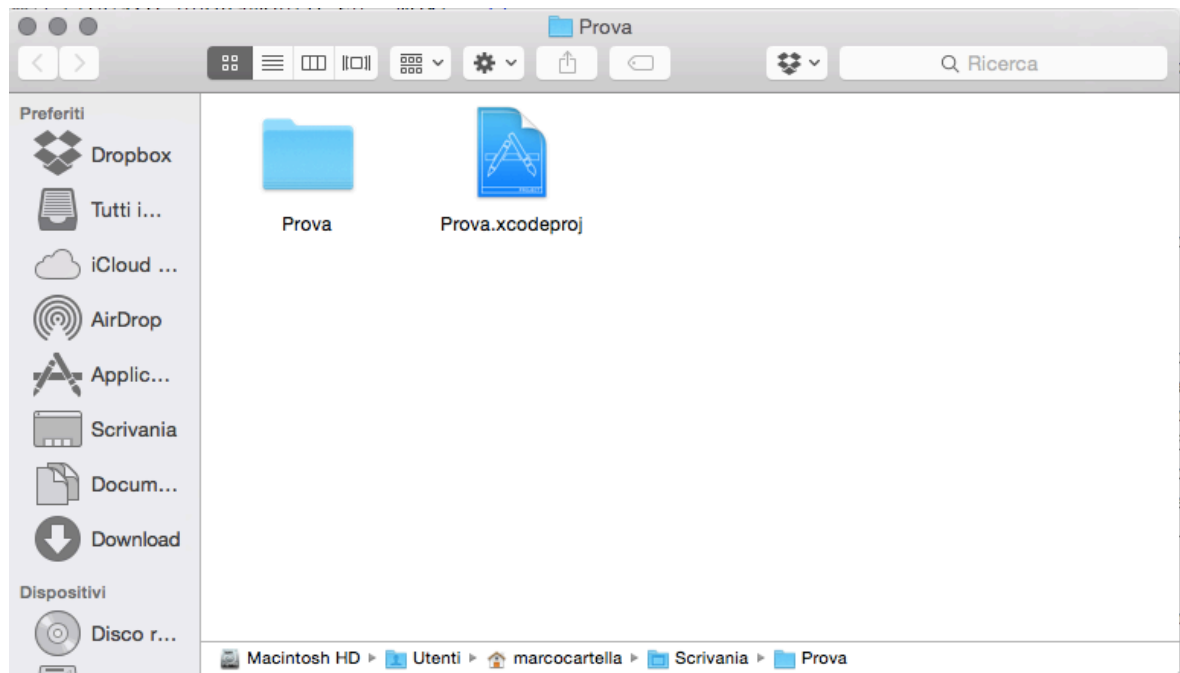


Figura 9 - Contenuto cartella di progetto

- Metodo utilizzabile solo se il progetto è stato creato col proprio MAC:
All'apertura di Xcode si apre una finestra di avvio che contiene alla destra la cronologia dei progetti e/o codici aperti (vedi Figura 4) con il MAC in questione. Quindi è possibile cercare il progetto nella lista e aprirlo direttamente da questa.
ATTENZIONE: Aprire il file con l'icona corrispondente a quella di file ".xcodeproj", vedi Figura 10.



Figura 10 - Icona di un file ".xcodeproj"

Una volta aperto il progetto, si rimanda al paragrafo 3.3 per una descrizione essenziale dell'interfaccia.

3.3 Interfaccia

Una volta creato o aperto il progetto apparirà l'interfaccia di lavoro (Figura 11) di Xcode, nella quale vi saranno numerose funzionalità, per il corso di Informatica molte di queste funzioni verranno ignorate.

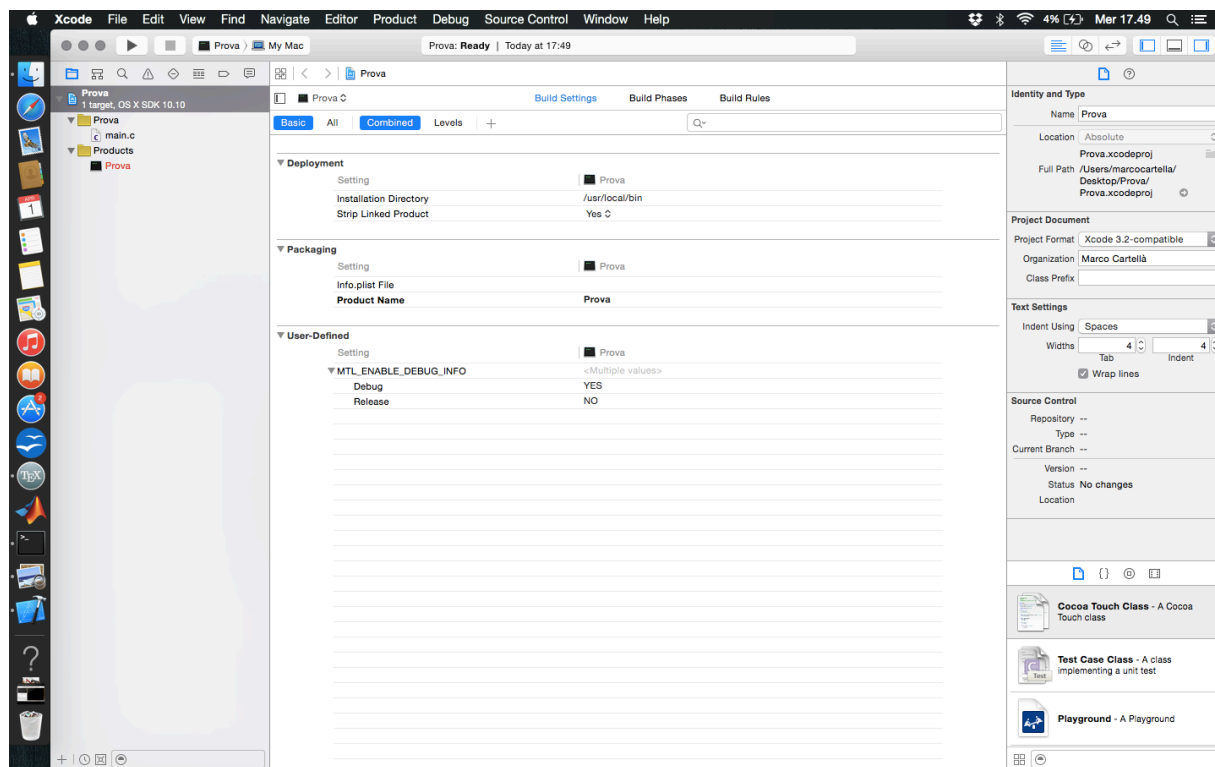


Figura 11 - Interfaccia di lavoro

Guardando superficialmente l'intero layout si nota una suddivisione dell'interfaccia di lavoro in tre zone:

- Una zona centrale, dove risiederà sempre il file o i file su cui si lavorerà.
- Una zona a sinistra, nella quale si trovano una serie di icone utili alla navigazione all'interno del codice e del progetto (Figura 12).



Figura 12 - Icone utili alla navigazione

Verranno trattate le icone più utili da sinistra verso destra:

La prima icona permette di navigare nel direttorio in cui risiedono i file di client o di implementazione con estensione ".c" e altri file come ad esempio quelli header (interfaccia) ".h" (che nel corso non verranno trattati). Cliccando una volta sul file verrà aperto nella zona centrale il file desiderato, mentre cliccando due volte si aprirà il file in una nuova finestra, si consiglia il singolo clic;

La quarta icona permette la visualizzazione e la navigazione dei warning o degli errori contenuti nel codice e rilevati in fase di compilazione;

La settima icona permette di trovare e navigare in modo efficiente tra tutti i breakpoint presenti nel progetto. Inoltre, è possibile disattivarli cliccando sulla icona blu facendola diventare un grigio. In Figura 13 è mostrata l'icona di attivazione/disattivazione dei breakpoint.

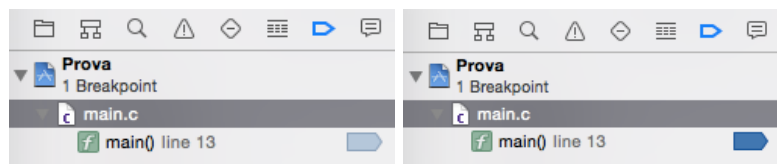


Figura 13 - A sinistra un breakpoint attivo, mentre a destra un breakpoint disattivo

- Una zona a destra, che non verrà utilizzata mai.

Vi è anche un'altra area dell'interfaccia che apparirà nella zona inferiore quando si eseguirà il codice oppure si vorrà fare il debug. Questa area è suddivisa essenzialmente in 2 parti: una parte sinistra che contiene l'elenco delle variabili e il loro contenuto utilizzate nel codice, e una a destra che corrisponde alla console per l'input e l'output del programma in esecuzione (vedi Figura 14).

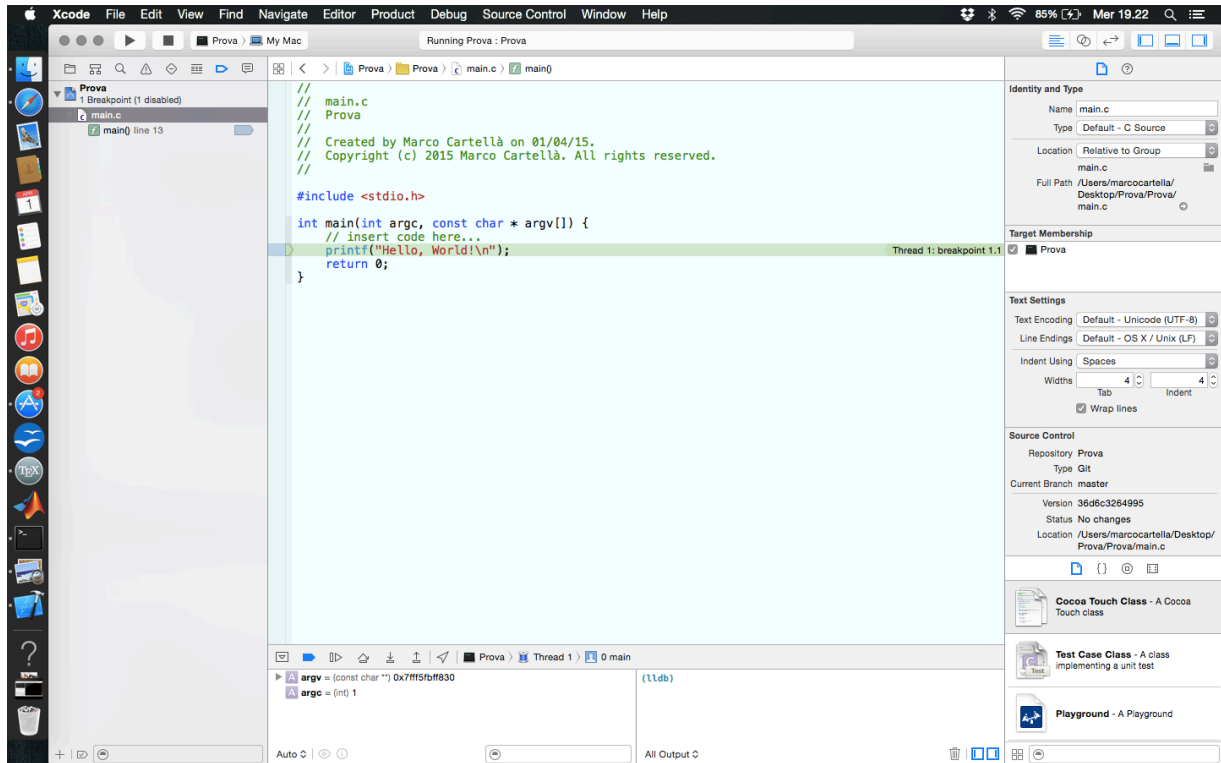


Figura 14 - Interfaccia dopo l'esecuzione di un programma

Vi è una ulteriore zona superiore che contiene una serie di icone utili per l'esecuzione e la modifica del layout di lavoro (Figura 15):



Figura 15 - Zona superiore

Verranno discusse a partire da sinistra verso destra:

- Le prime 3 icone sono le classiche icone di OS X che permettono la chiusura, l'espansione o la riduzione della finestra. Quando il progetto viene chiuso il tutto viene salvato automaticamente.
- "Run": è il bottone con all'interno un triangolo permette di fare il build e eseguire il codice a partire dal main();
- "Stop": è il bottone con al suo interno un quadrato ferma l'esecuzione di un programma;
- A destra vi è il bottone utile a inserire i parametri da linea di comando;
- Centralmente vi è una vidiaata contenente lo stato dell'esecuzione, nell'esempio in Figura 14 il codice è in esecuzione (running).
- Successivamente ci sono 3 bottoni che non avranno utilità particolari per l'utilizzo del CORSO;
- Per concludere ci sono altre tre bottoni che permettono di mostrare o nascondere le varie aree dell'interfaccia di lavoro discusse precedentemente.

3.4 Debug

Per fare partire il debug con Xcode bisogna eseguire due passaggi:

- Inserire un breakpoint: Per inserirlo *clickare nella zona interposta tra l'area a sinistra e quella centrale* (vedi la descrizione dell'interfaccia nel paragrafo 3.3). Comparirà una icona blu simile a un segnalibro;
- Eseguire il codice: *Cliccare il bottone "Run"* in alto a sinistra.

Una volta cliccato sul pulsante "Run" l'esecuzione si metterà in pausa dove è stato inserito il breakpoint, si noterà dal fatto che la riga sarà evidenziata di verde (vedi Figura 16).

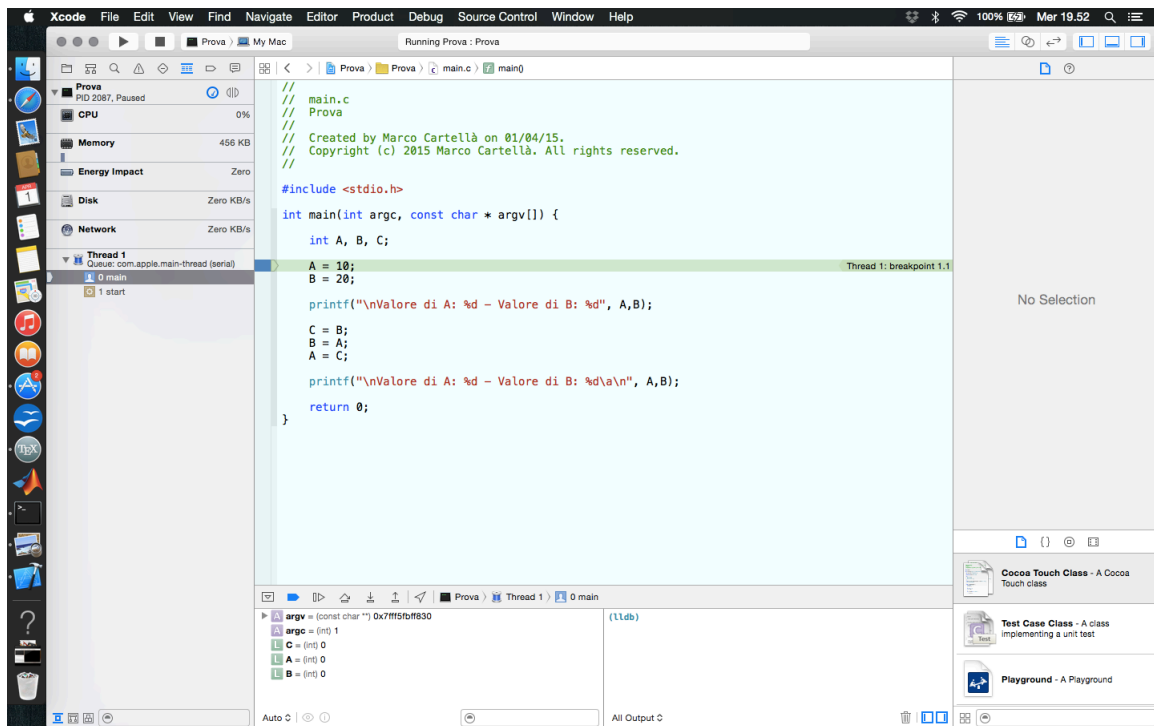


Figura 16 - Esecuzione ferma al breakpoint

Nella parte a sinistra della zona inferiore è possibile controllare in tutto e per tutto il debug: una barra per il controllo del flusso di esecuzione e una zona dedicata alle variabili dichiarate nel programma (Figura 17).

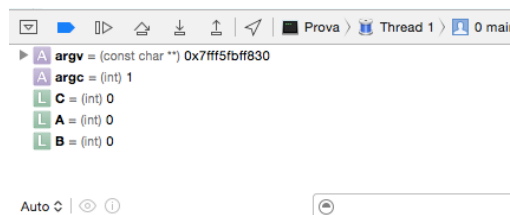




Figura 17 - Interfaccia per il debug

In questa zona ci sono una serie di bottoni che permettono di controllare il flusso di esecuzione durante il debug, verranno discussi a partire da sinistra verso destra:

- Nascondere l'area di debug;
- Attivare e disattivare tutti i breakpoint;
- Continuare l'esecuzione del programma. In caso di presenza di un breakpoint successivo l'esecuzione si fermerà ad esso;
- "Step Over": andare all'istruzione successiva senza entrare nella funzione;


-  "Step in": andare all'istruzione successiva entrando all'interno della funzione (questa funzionalità è supportata solo per le funzioni che hanno una implementazione visibile).
-  "Step out": uscire dalla funzione e tornare al chiamante, dopo che la funzione in questione è stata eseguita.


In questa area è possibile inoltre inserire variabili non presenti per visualizzarne il contenuto, un esempio può essere un vettore allocato dinamicamente. Per fare questa operazione:

- Cliccare con il pulsante destro (o con due dita) nell'area bianca dell'area delle variabili;
- Cliccare su "Add Expression...";
- Inserire il nome della variabile.


Se si hanno problemi con il debug forse le FAQ possono risolvere il problema. Clicca [qui](#).

3.5 Linea di comando

Per inserire da linea di comando i parametri da passare in ingresso bisogna modificare lo schema del progetto. Questa operazione viene effettuata cliccando sul nome del progetto nella barra in alto (in Figura 15) con icona  e andare su "Edit Scheme...". A questo punto si aprirà una nuova finestra contenente le varie preferenze sullo schema del progetto, si consiglia di non modificare nulla. Per inserire i parametri:

- Andare nella sezione "Run - Debug" (selezionabile nell'area a sinistra);
- Selezionare la scheda "Arguments" (in alto);
- In "Arguments Passed On Launch" clicare sull'icona  ;
- Inserire le stringhe da passare come parametro separate da spazio o, se i parametri fossero composti da più parole inserire i parametri tra doppie virgolette separati poi da spazi. Ad esempio:

"file Input.txt" "File Output.txt" 1000 40

In Figura 18 un esempio di come deve essere modificato lo schema. Vi è inoltre la possibilità di eliminare un parametro selezionandolo col flag e cliccando sull'icona .

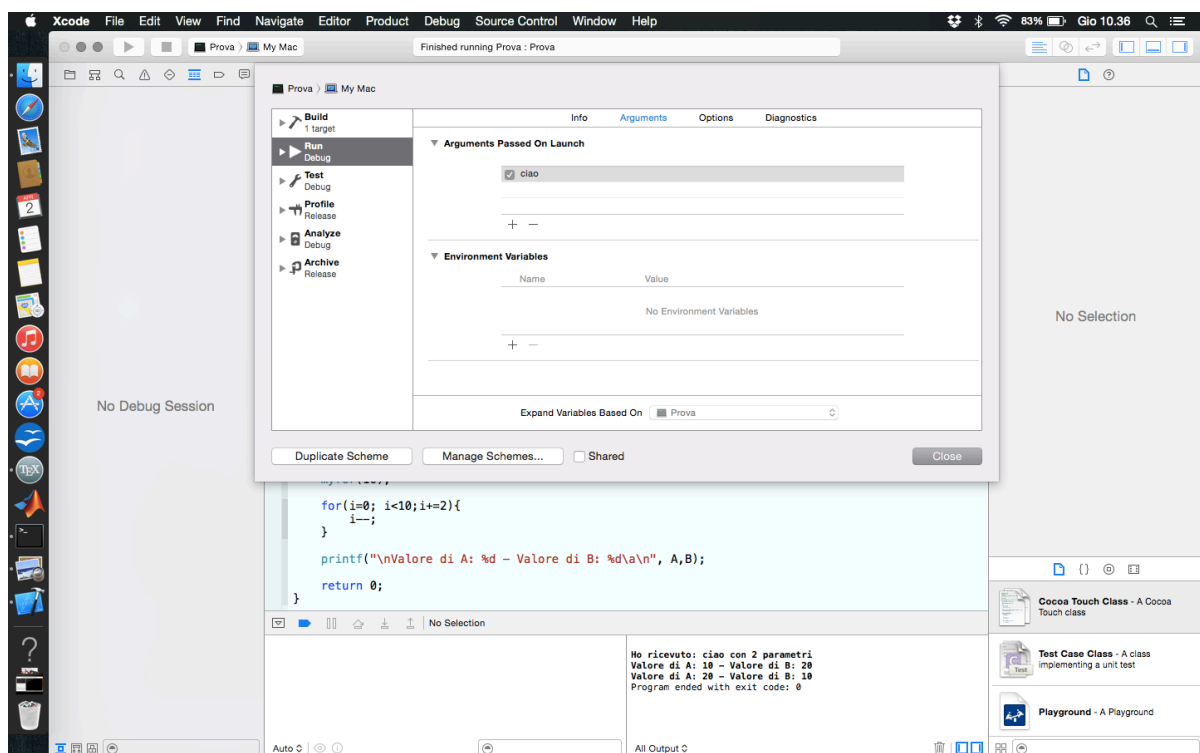



Figura 18 - Preferenze dello schema, zona parametri linea di comando

3.6 Eseguibile e linea di comando

Una volta scritto e compilato il codice scritto viene creato un file binario. Su Xcode il file binario non risiede di default nella stessa directory del progetto. Nonostante ciò viene comunque mostrato nella cartella *Products* su Xcode, come mostra la Figura 11.

Se si desidera utilizzare il file binario è possibile cambiare il percorso di destinazione di questo eseguendo questi passaggi:

- Andare nel menù di Xcode cliccando in alto a sinistra nel tab strumenti (affianco la mela) su "Xcode" ;
- Cliccare su "Preferences...";
- A questo punto apparirà la finestra delle preferenze di Xcode: andare al tab "Locations" (la Figura 19 mostra la finestra);

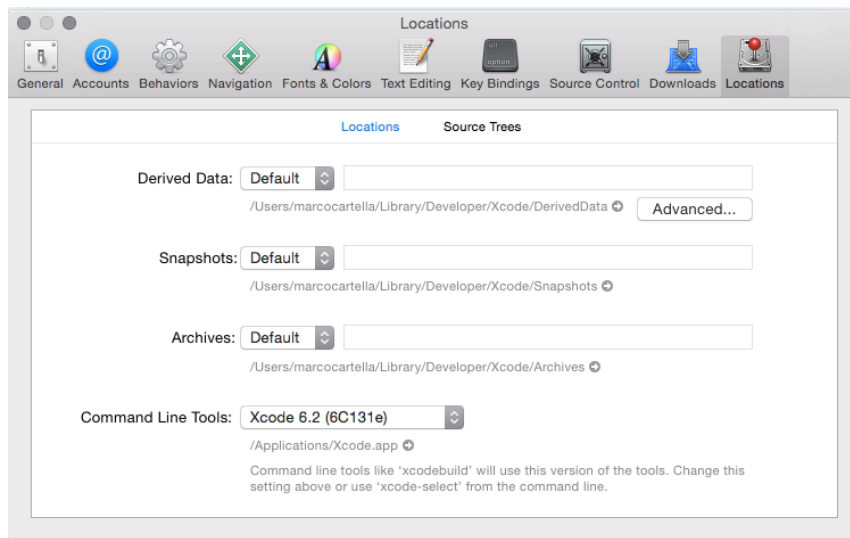



Figura 19 - Preferenze Xcode nel tab "Locations"

- Cliccare sul bottone ;
- Spuntare "Custom";
- Selezionare dal menù a tendina "Relative to Workspace".

In Figura 20 è mostrato come deve essere compilata la sezione "Advanced".

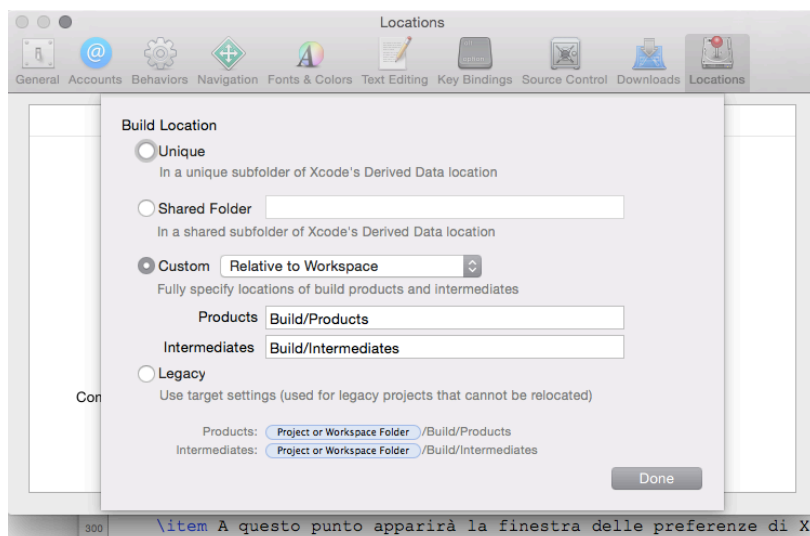


Figura 20 - Finestra "Advanced" dopo le modifiche

Successivamente dopo una ulteriore compilazione cliccando il bottone "Run" è possibile trovare il file binario nella cartella del progetto nel percorso

<percorso progetto>/Build/Products/Debug.

Una volta ottenuto il file binario è possibile eseguirlo da linea di comando. Quindi, per prima cosa, aprire il terminale (icona in Figura 21). Per aprirlo premere il tasto F4, una volta premuto si aprirà il launchpad (vedi Figura 22), dove trovate tutte le vostre applicazioni installate. Successivamente vi sarà fra le pagine del launchpad una cartella denominata "Altro", all'interno della quale troverete il terminale (il contenuto è mostrato in Figura 23).



Figura 21 - Icona Terminale

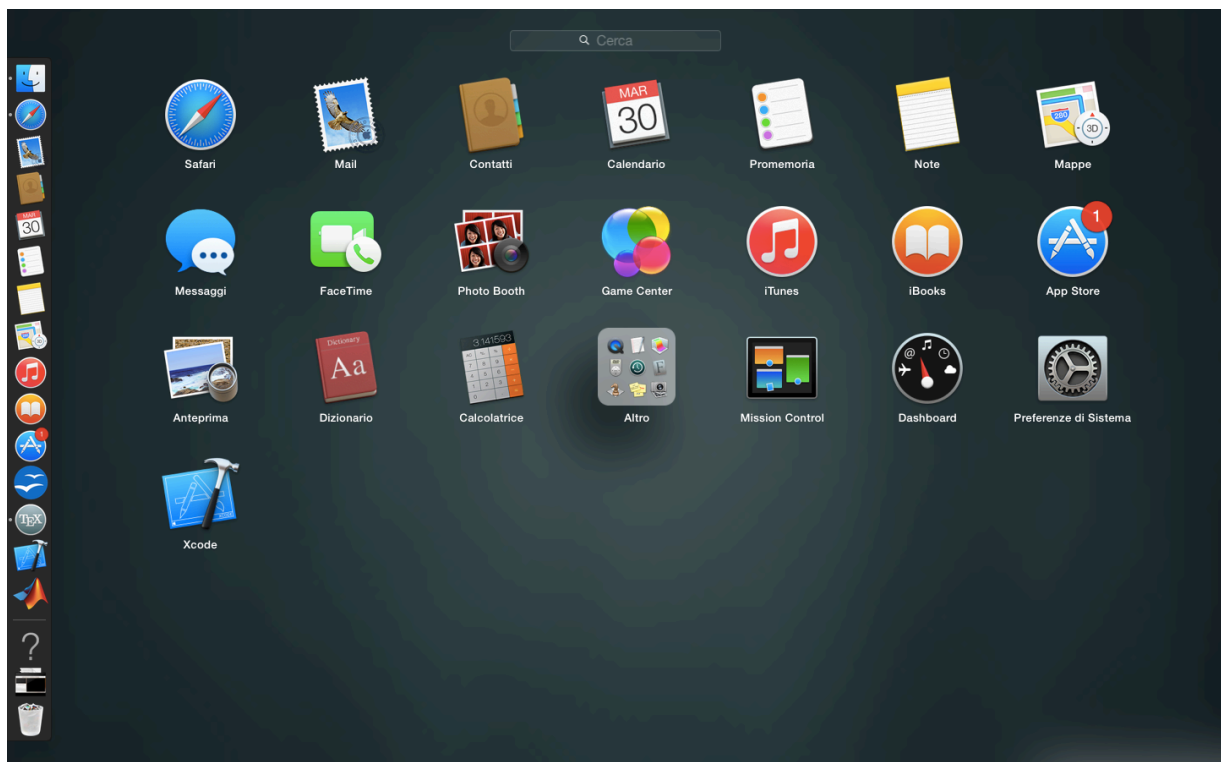


Figura 22 - Launchpad



Figura 23 - Contenuto cartella "Altro"

Una volta aperto, si aprirà il terminale bash (Figura 24)

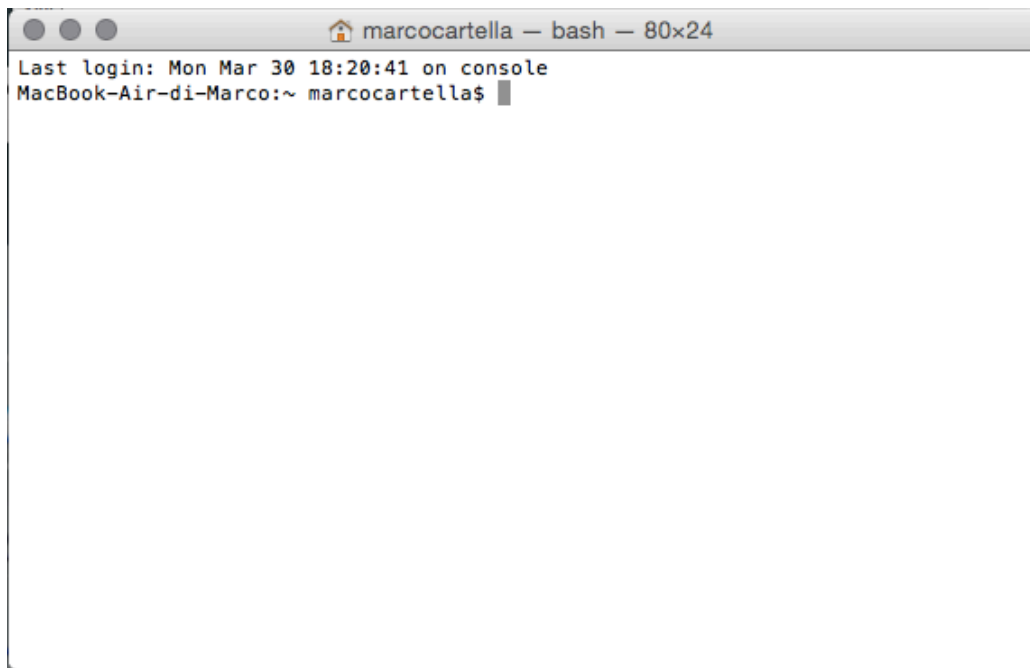


Figura 24 - Terminale bash

Una volta aperto il terminale *muoversi all'interno della directory* del file binario con il seguente comando (sostituire <percorso progetto> con il percorso reale del proprio progetto):

```
cd <percorso progetto>/Build/Products/Debug
```

Successivamente *eseguire il binario*:

```
./<nome binario> <parametro1> <parametro2> ... <parametroN>
```

In Figura 25 è mostrato un esempio.

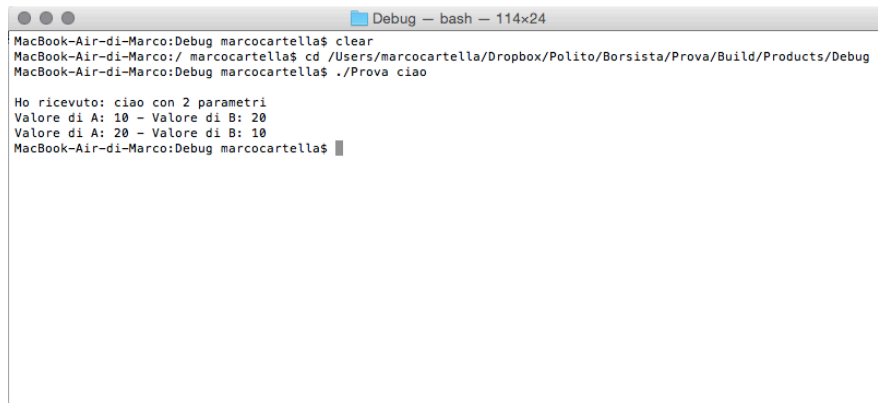


Figura 25 - Esempio di apertura di un file eseguibile binario da terminale

3.7 Lettura e scrittura file

Questa sezione mostrerà come creare un file .txt con Mac OS X con l'utilizzo del terminale, azione necessaria poiché Mac OS X non ha software nativi per la creazione di questo tipo di file. Inoltre nella seconda parte si mostrerà come cambiare il direttorio di lavoro per la lettura o per la scrittura di file, procedura necessaria affinché il file sia trovato.

3.7.1 Creazione di un file .txt

L'operazione è relativamente semplice, seguire i passaggi seguenti:

- Aprire il terminale (clicca [qui](#) per andare alla sezione che mostra come farlo);
- Utilizzare il comando "cd" (acronimo di change directory) che permetterà di muoversi in un direttorio specifico. Quindi, ad esempio:

```
cd Desktop/
```

- Utilizzare il comando "touch" che creerà un nuovo file vuoto:

```
touch <nomefile>.txt
```

è importante mettere l'estensione (.txt) del file. Dopo questa operazione dovreste trovare il file creato sul desktop e potrete copiarlo nella cartella del progetto Xcode.

In Figura 26 vi è la sequenza di comandi inseriti su terminale. Affinché il comando "cd Desktop/" funzioni è necessario che il terminale sia stato appena aperto e non si sia utilizzato precedentemente il comando cd. Mentre in Figura 27 è mostrata l'icona che dovrebbe apparire sul desktop dopo l'esecuzione del comando touch da terminale.

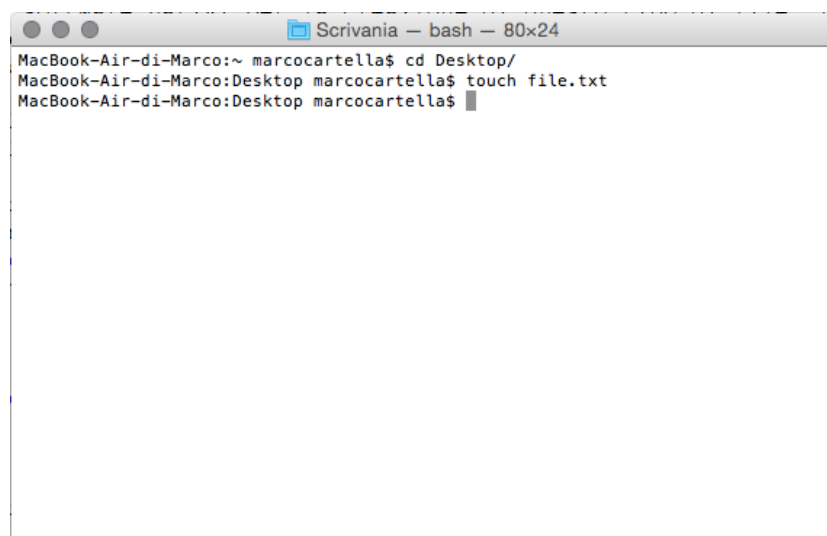


Figura 26 - Creazione file con comando touch da terminale

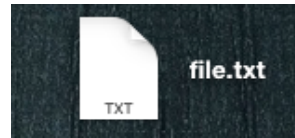




Figura 27 - Icona del file .txt creato

3.7.2 Direttorio di lavoro

Con le nuove versioni di Xcode il direttorio di lavoro non è più quello del progetto (come avviene con Code::Blocks) ma un direttorio separato, interno all'ambiente di lavoro di Xcode. Quindi di norma i file su cui si vuole leggere non verranno trovati, oppure il file su cui avete scritto non si troverà nella cartella del progetto. Per cambiare questo comportamento *ogni qual volta si vuole scrivere o leggere un file*, quindi ad ogni progetto, bisogna modificare il direttorio di lavoro. Per fare questo, seguire i passaggi elencati:

- Cliccare sull'icona, in alto a sinistra,  e andare su "Edit Scheme...";
- Andare nella sezione "Run - Debug" (selezionabile nell'area a sinistra);
- Selezionare la scheda "Options". In Figura 28 è mostrato come deve essere impostato il pannello "Options";
- Mettere la spunta su "use custom working directory";
- Cliccare sull'icona ;
- Andare nella cartella dentro la quale vi è il file da leggere o dove si vuole scrivere il file. In Figura 29 è mostrato un esempio.
- Cliccare su "Choose";
- Cliccare su "Close".

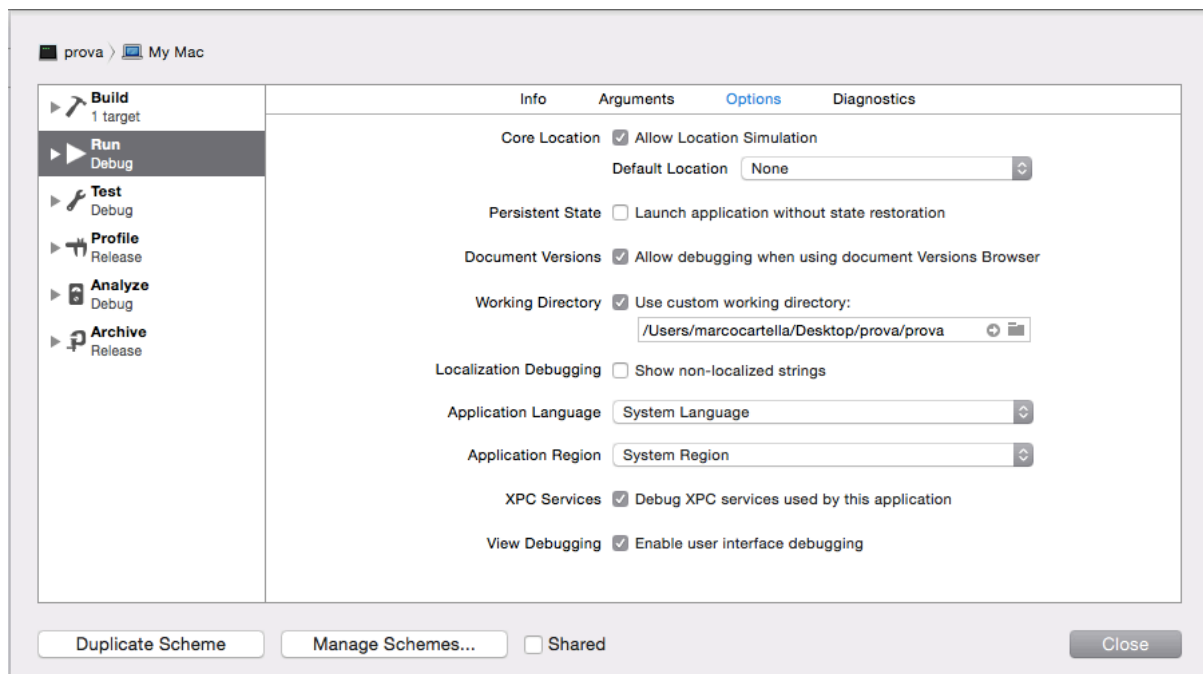


Figura 28 - Pannello Options

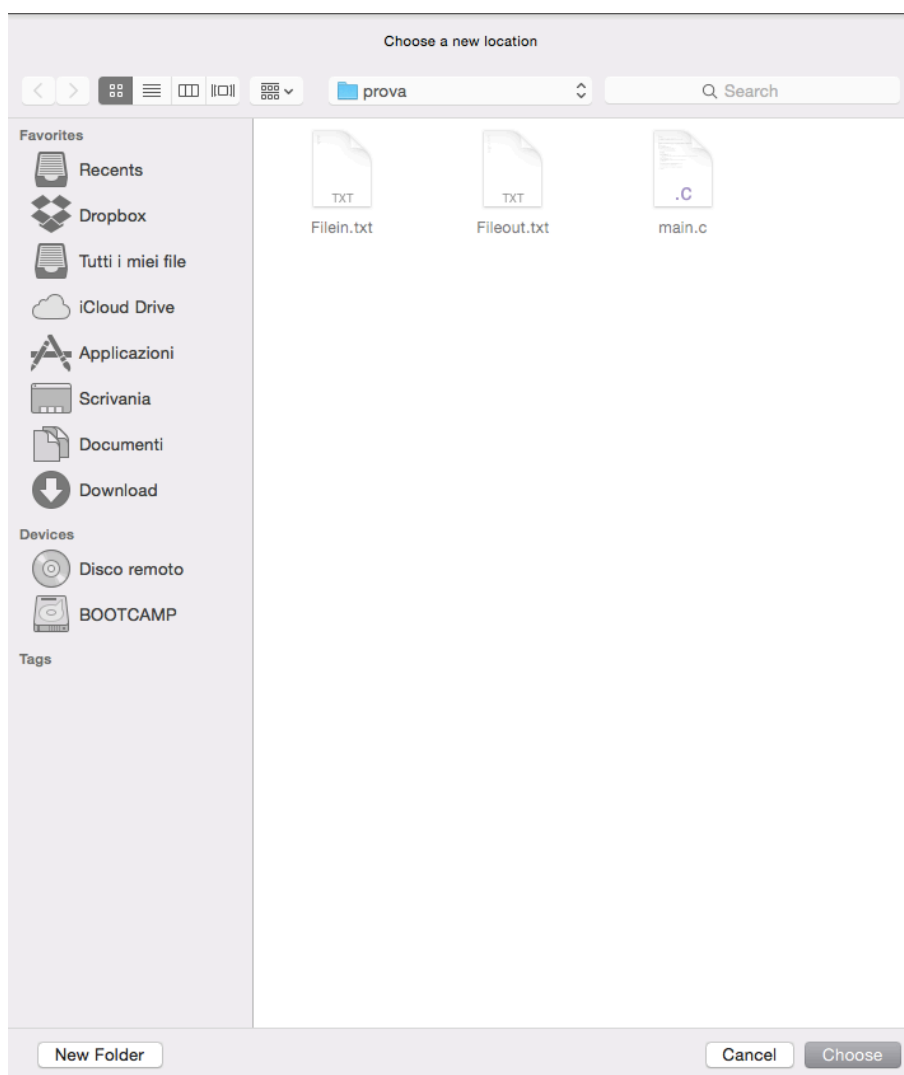


Figura 29 - Scelta del direttorio di lavoro

4 Dopo il corso

Alla fine del corso e una volta conseguito l'esame, vi saranno due pensieri: chi vorrà disinstallare Xcode e chi lo vorrà mantenere installato per utilizzi futuri. La disinstallazione di questa applicazione non è banale come può sembrare poiché la sua installazione implica l'installazione di numerosi tool. Inoltre, pur essendo una normale applicazione per Mac, non può essere rimossa con il solito trascinamento nel cestino; dobbiamo dunque seguire un determinato procedimento.

Si consiglia la disinstallazione, in caso di inutilizzo, probabilmente utilizzate il vostro Mac per lavori di bassa-media entità e per il quale non è richiesta una piattaforma di sviluppo dedicata al lavoro di sviluppatori e/o programmatori. Disinstallando XCode quindi, recuperereste diversi gigabyte di spazio sul vostro disco rigido così da poterli dedicare ai vostri lavori, dati o progetti preferiti.

4.1 Disinstallazione

4.1.1 Procedura consigliata

Aprire il terminale (vedi paragrafo 3.6 per l'apertura del terminale).

All'interno del terminale inserite il seguente comando:

```
sudo /Developer/Library/uninstall-devtools --mode=all
```

Successivamente all'inserimento di questo comando vi verrà richiesto di inserire la password utilizzata per lo user del MAC, inseritela e premete invio sulla tastiera sulla tastiera (i caratteri inseriti non verranno mostrati, inseriteli senza pensarci). Fatto questo, avrete dunque avviato il processo di rimozione totale di XCode che richiederà dai 2 ai 5 minuti circa, a seconda del vostro sistema.

Una volta completato il processo di rimozione dell'applicazione, è consigliabile cancellare anche le directory native di Xcode situate nella sezione "Applicazioni" del vostro Mac; così facendo libererete circa 1,7 Gb di spazio dal vostro disco rigido.

4.1.2 Procedura alternativa

NOTA IMPORTANTE: Seguire questa procedura solo nel caso in cui la prima procedura non funzioni.

Può succedere che durante l'installazione Xcode non si crei la cartella Developer, ve ne accorgete nel caso in cui i comandi sopra enunciati non vengono eseguiti con successo poiché il terminale non trova la directory.

In questo caso vi basterà eseguire 2 semplici comandi sempre dall'applicazione "Terminale":

- Muoversi nella cartella "Applications" inserendo il seguente comando da terminale:

```
cd /Applications/
```

- Rimuovere l'applicazione inserendo il seguente comando da terminale:

```
sudo rm -rf Xcode.app
```

Anche in questo caso richiederà la password, inserire quindi la password, e attendete qualche secondo che la disinstallazione avvenga.

4.2 Rimozioni parti poco utili di Xcode

Se invece volete rimuovere solo una parte che non vi interessa di XCode alleggerendo quindi il programma e il vostro sistema, il procedimento è lo stesso del paragrafo 4.1.1 ma con altri comandi da inserire nel terminale:

- Rimuovere XCode "Dev-Tool":
Una cosa che può risultarvi poco utile o comunque che non vi serve è la directory di sviluppo avanzata appunto la "Dev-Tool"; ecco il codice terminale per rimuoverla:

```
sudo /Developer/Library/uninstall-devtools --mode=unixdev
```

- Rimuovere XCode “Directory di sviluppo e contenuti”:

La stessa cosa vale per la Directory di sviluppo e contenuti e per la directory riguardante il supporto del sistema:

```
sudo /Developer/Library/uninstall-devtools --mode=xcodedir
```

- Rimuovere XCode “Supporto di Sistema”:

```
sudo /Developer/Library/uninstall-devtools --mode=systemsupport
```

5 Frequently Asked Questions - FAQ

5.1 Problemi con il debug

5.1.1 Vista assembler

Capita che di default quando si va a debuggare il codice appaia, invece del codice C, un codice che può sembrare satanico (vedi Figura 30). Non bisogna preoccuparsi, è il codice assembler (o assembly). Il compilatore lo genera in fase di compilazione a partire dal codice C. Per rimuovere questa opzione seguire i passaggi elencati:

- Cliccare nella barra degli strumenti in alto su **Debug**;
- Andare su "Debug Workflow";
- Cliccare su "Always Show Disassembly" per eliminare la spunta, vedi Figura 31.

Ora si potrà debuggare il codice che avrete scritto in modo "umano".

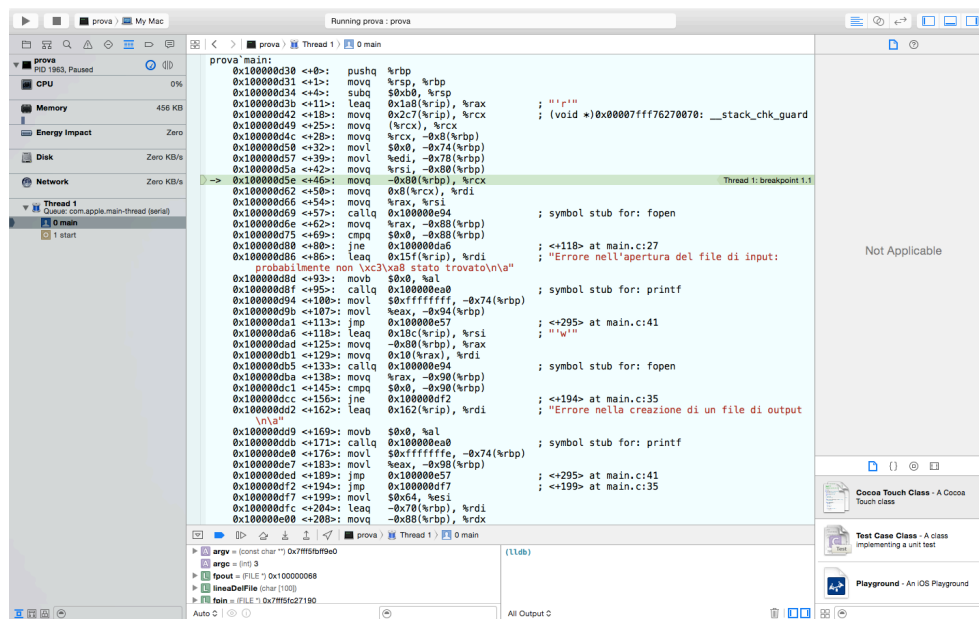


Figura 30 - Codice assembler durante il debug

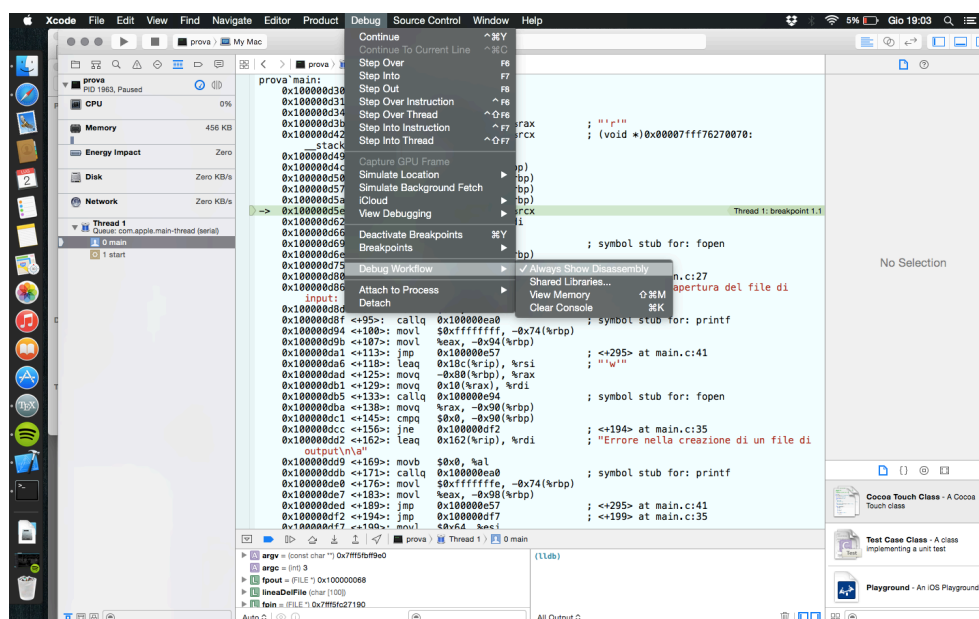


Figura 31 - Come eliminare il problema

5.2 Funzione System()

Il comando `system()` serve per eseguire comandi da linea di comando. Il terminale Mac non è quello di Windows e, ad esempio, il comando `CLS` funziona solo su linea di comando Windows e non da quello di MAC OS X. Tuttavia si potrebbero utilizzare i corrispondenti comandi POSIX (ad esempio il corrispondente di `CLS` è `"clear"`), in quanto su MAC OS X il terminale è quello POSIX. Quindi, in teoria, ma solo in teoria purtroppo, il comando `system("clear")` dovrebbe funzionare. Ma se si prova a eseguire la console mostrerà il messaggio `"TERM environment variable not set"`. Il motivo di tale risposta è che la console che utilizza Xcode non è neanche POSIX ma solo una console di output proprietaria Apple. `TERM` è una variabile d'ambiente che informa che tipo di terminale si sta utilizzando e sulla console di Xcode non è settata. Rimanendo nell'esempio precedente, tuttora non esistono comandi in grado di effettuare la pulizia della schermata.

In ogni caso, il comando `system` non è indispensabile per il corso. Unica funzione che forse si vedrà utilizzare più spesso sarà `system("exit")` o simili che possono essere tranquillamente sostituite da semplici `"return n"` dove n è un numero diverso da 0 (ad esempio 1 o -1) che può indicare l'uscita con errore del programma.

Riferimenti bibliografici

- [1] Apple inc., *Xcode on the Mac App Store*, 10 Novembre, 2014.
- [2] Apple inc., *Mac Dev Center*, 31 Luglio, 2011.
- [3] Prince McLean, *iOS Developer Library*, 20 Giugno, 2008.
- [4] Apple inc., *iOS Developer Library*