



Progetto API

Anno 2019-2020



Agenda

1. Canali di comunicazione
2. Incontri
3. Tools
4. Verificatore
5. Review della consegna
6. A.M.A. (Ask Me Anything)



Tutor

- Luca Collini: luca.collini@mail.polimi.it
- Luca Terracciano: luca.terracciano@mail.polimi.it



Dove contattarci

- Questioni burocratiche:
 - Contattare il Professore
- Comprensione della traccia:
 - Scrivere sull'aula virtuale (se è una domanda che ritenete possa essere di interesse generale)
 - Mail
- Problemi di implementazione/tecnici:
 - Mail
 - Sessioni di tutoring
- Problemi con il verificatore:
 - Scrivere sulla piattaforma



Incontri

Proposte Giugno/Luglio:

- 27 Giugno
- 3 Luglio
- 23 Luglio

Uno o due incontri a Settembre da definire



Tools

Tipo di problema	Tool
Crash	GDB
Output non corretto	GDB/Valgrind
Problemi di performance	Valgrind

[Vademecum dei tools](#)



GDB - GNU Debugger

GDB permette di vedere cosa succede all'interno del vostro programma C/C++.

Offre la possibilità di:

- Definire dei breakpoint nel codice
- Far procedere l'esecuzione del codice linea per linea
- Vedere il contenuto delle variabili

Utile nel momento in cui si verificano dei crash o anomalie nel codice.

Uso: consultare le sezione GDB in [Vademecum dei tools](#)



Valgrind

Valgrind mette a disposizione molteplici tools che possono essere usati per analizzare il codice.

I più utilizzati sono:

- Memcheck
- Massif
- Callgrind



Massif

Massif registra l'utilizzo dello heap nel tempo.

Permette di rilevare picchi di memoria indesiderati all'interno di una certa funzione nel codice.

Uso: `valgrind --tool=massif --massif-out-file=output_file ./program_to_run`

Per controllare il risultato:

- `ms_print output_file`
- `massif_visualizer output_file`



Memcheck

Memcheck è un rilevatore di errori di memoria. Rileva errori comuni nella programmazione in C:

- Accessi alla memoria ad indirizzi non validi, come sopra o sotto heap/stack o indirizzi che sono stati liberati
- Utilizzo di valori non inizializzati
- Errori nella deallocazione della memoria
- Memory leaks

Sono tipicamente errori che non fanno crashare il programma ma causano output scorretti o utilizzo della memoria spropositato.

Uso: `valgrind --leak-check=full --show-leak-kinds=all --track-origins=yes ./program_to_run`

<https://valgrind.org/docs/manual/mc-manual.html#mc-manual.options>



Callgrind

Callgrind è un tool per la profilazione che registra le chiamate alle funzioni del programma creando un grafo delle chiamate.

Possiamo poi usare kcachegrind per visualizzare l'output di callgrind e capire quali sono i colli di bottiglia del nostro codice.

Uso:

```
valgrind --tool=callgrind --dump-instr=yes --callgrind-out-file=callgrind.out ./program_to_run
```

Poi per controllare il risultato:

```
kcachegrind callgrind.out
```



Verificatore

<https://dum-e.deib.polimi.it/>



La consegna

Il progetto consiste nell'implementare un semplice editor di testi con interfaccia a comandi testuali.

L'editor considera un documento come una sequenza di righe, di dimensione arbitraria (ognuna con al massimo 1024 caratteri), numerate a partire da uno.



Comandi

I comandi sono costituiti da un'unica lettera, preceduta opzionalmente da uno o due numeri interi.

- c - change
- d - delete
- p - print
- u - undo
- r - redo
- q - quit



change

(ind1,ind2)c ...

Cambia le righe del documento da *ind1* a *ind2* incluse

N.B. *ind1* deve essere presente nel documento o essere il primo indirizzo dopo l'ultima riga

```
1,2c
```

```
prima riga
```

```
seconda riga
```

```
.
```



delete

(ind1,ind2)d

Cancella le righe del documento da *ind1* a *ind2* inclusi spostando verso l'alto le righe successive a quella di indirizzo *ind2*.

La cancellazione di righe inesistenti non ha alcun effetto.



print

$(ind1, ind2)p$

Stampa le righe comprese tra $ind1$ e $ind2$.

Le righe non esistenti tra i due indici
vengono sostituite con una riga
contenente un punto.

```
prima riga  
nuova seconda riga  
terza riga
```

$(1,5)p$

```
prima riga  
nuova seconda riga  
terza riga
```

.

.



undo

(numero)u

Annulla un numero di comandi di tipo:

- change
- delete

Se il numero di comandi di cui effettuare l'annullamento `e superiore a quello dei comandi eseguiti, vengono annullati tutti i passi.

Se dopo un undo si effettua una modifica al testo si perde la possibilità di eseguire un redo dei comandi successivi



redo

(numero)r

Annulla un numero di undo fatti precedentemente

Nel caso in cui il numero di comandi di cui effettuare il redo è superiore a quelli annullati correntemente, viene effettuato il numero massimo di redo possibili.



quit

q

Termina l'esecuzione dell'editor.



Assunzioni

- All'editor vengono dati comandi corretti
- A volte i comandi sono corretti ma non hanno effetto
 - delete di righe non esistenti
 - redo di un numero di comandi maggiore di quelli presenti nello storico

A.M.A.

Ask Me Anything

