Esempio GDB

C debugger

Giuseppe Prencipe
Massimo Torquati

Dip. Informatica
Univ. Pisa

```
#include <stdio.h>
#include "string.h"

int main(int argc, char* argv[]) {
	int i, j, k;
	int x[1000];

	for(i = 0; i < 10000; ++i){
		x[i] = i;
	}

	printf("Enter integer in 0..9999: ");
	scanf("%d", k);

	tester(x, k);
}

int tester(int* c, int k) {
	printf("x[%d] = %d\n", k, c[k]);
}
```

GDB

- Consideriamo il file di esempio (gdbExample.c)
- Compiliamo
  - $ gcc **-g** gdbExample.c
- Eseguiamo
  - ./a.out

GDB

- Consideriamo il file di esempio (gdbExample.c)
- Compiliamo
  - $ gcc **-g** gdbExample.c
- Eseguiamo
  - ./a.out
  - Va in crash….

GDB

- Come mai?
- Proviamo ad usare il debugger
  - gdb a.out
    - Poi ‘run’ per eseguire il programma all’interno del gdb

GDB

- Come mai?
- Proviamo ad usare il debugger
  - gdb a.out
    - Poi ‘run’ per eseguire il programma all’interno del gdb
      - Errore in linea 9
        - Qualcosa non va con ‘i’
          - ‘print i’ — ‘print x[10]’ — ‘print x[i]’

GDB

- Proviamo ad usare il debugger
    - gdb a.out
        - Poi ‘run’ per eseguire il programma all’interno del gdb
- Errore in linea 9
    - Qualcosa non va con ‘i’
        - ‘print i’ — ‘print x[10]’ — ‘print x[i]’
        - Qualcosa non va alla posizione i
        - Stampiamo il codice
            - ‘list’

GDB

- Proviamo ad usare il debugger
  - gdb a.out
    - Poi ‘run’ per eseguire il programma all’interno del gdb
      - Errore in linea 9
        - Qualcosa non va con ‘i’
          - ‘print i’ — ‘print x[10]’ — ‘print x[i]’
          - Qualcosa non va alla posizione i
          - Stampiamo il codice
            - ‘list’
              - indice i troppo grande! —> sistemiamo il codice….

GDB

- Dopo aver sistemato il codice (in gdbExample_OK.c), eseguiamo nuovamente, e avviamo gdb
    - gdb a.out
        - Poi ‘run’ per eseguire il programma all’interno del gdb
- Impostiamo un *breakpoint* alla linea 8
    - ‘break gdbExample_OK.c:8’
    - ‘run’
    - usiamo ‘step’ per eseguire un passo alla volta
        - ‘print i’ per verificare a che punto siamo

GDB

- usiamo ‘step’ per eseguire un passo alla volta
    - ‘print i’ per verificare a che punto siamo
        - Troppo lungo….inseriamo secondo breakpoint alla linea 12, e poi ‘continue’ per arrivare al secondo breakpoint appena impostato
        - Altro ‘continue’ per arrivare fino alla fine (altro *segmentation fault*)

GDB

- usiamo ‘step’ per eseguire un passo alla volta
    - ‘print i’ per verificare a che punto siamo
        - Troppo lungo….inseriamo secondo breakpoint alla linea 12, e poi ‘continue’ per arrivare al secondo breakpoint appena impostato
    - Altro ‘continue’ per arrivare fino alla fine (altro *segmentation fault*)
        - Invocando ‘bt’ viene stampato lo stack, e ci rendiamo conto meglio di cosa possa essere successo….

GDB

- usiamo ‘step’ per eseguire un passo alla volta
  - ‘print i’ per verificare a che punto siamo
    - Troppo lungo….inseriamo secondo breakpoint alla linea 12, e poi ‘continue’ per arrivare al secondo breakpoint appena impostato
  - Altro ‘continue’ per arrivare fino alla fine (altro *segmentation fault*)
    - Invocando ‘bt’ viene stampato lo stack, e ci rendiamo conto meglio di cosa possa essere successo….
    - Problema con la scanf, che prende int e non *int (e d’altronde era anche tra i warnings!!!!)