

# **gdb: manuale di rapida consultazione**

## **Argomenti trattati:**

- Settare bp
- Vedere bp settati
- Cancellare bp
- Disabilitare bp
- Riabilitare bp
- Settare wp
- Visualizzare wp
- Condizioni per i bp/wp
- Modificare/Aggiungere la condizione
- Cancellare la condizione
- Eseguire lo stop del programma solo dopo m volte che si e' verificato il bp/wp
- Eseguire comandi in automatico quando si verifica lo stop per un bp
- Continuare l'esecuzione del programma
- Vedere il disassemblato
- Assegnare il valore ad una variabile
- Continuare ad un indirizzo diverso
- Ritornare da una funzione
- Chiamare funzioni
- Stampare il valore di una espressione
- Visualizzare il contenuto delle locazioni di memoria
- Settare una visualizzazione automatica
- Visualizzare le visualizzazioni automatiche settate
- Disabilitare le visualizzazioni automatiche settate
- Riabilitare le visualizzazioni automatiche settate
- Cancellare le visualizzazioni automatiche settate
- Vedere tutti i registri e i loro valori
- Esaminare la symbol-table

## **Legenda:**

... valgono tutte le opzioni valide per il comando base  
[ ] parametri opzionali  
bp breakpoint  
wp watchpoint

## *Suggerimento:*

Create l'alias `gdb="gdb -q"` in modo che gdb parta senza visualizzare le info iniziali.

## **Settare bp:**

Comando **break** (abbreviato b)

**break** nome\_funzione

bp sulla funzione (solo se sono presenti le info di debug, o se avete usato labelz ;)

**break** +offset

**break** -offset

bp offset linee dopo/prima l'istruzione corrente

**break** \*indirizzo

bp all'istruzione a indirizzo

**break** (niente)

bp all'istruzione successiva a quella corrente

bp condizionati:

**break** ... if condizione

stop solo se e' valutata VERA la condizione

bp particolari:

**tbreak** ...

come break ma valido solo per uno stop, poi viene cancellato automaticamente

**Vedere bp settati:**

**info breakpoints** [n]

visualizza tutti i bp (o il bp numero n se indicato)

Indicazioni visualizzate:

Numero\_bp | Tipo (bp/wp) | Disposizioni | Abilitato(y)/Disabilitato(n) | Indirizzo\_bp |

Posizione\_nel\_sorgente

**Cancellare bp:**

Comando **delete** (abbreviato d)

**delete** [n]

cancella tutti i bp (o il bp numero n se indicato)

Comando **clear** (se si da il comando appena dopo che il programma si e' appena fermato ad un bp si cancella il bp stesso)

**clear** [nome\_funzione]

cancella il bp sulla prossima istruzione (o il bp su nome\_funzione se indicato)

**Disabilitare bp:**

Comando **disable** (abbreviato dis)

**disable** [n]

disabilita tutti i bp (o il bp numero n se indicato)

### **Riabilitare bp:**

Comando **enable**

**enable** [n]

riabilita tutti i bp (o il bp numero n se indicato)

**enable** once [n]

riabilita tutti i bp (o il bp numero n se indicato) ma al primo stop viene disabilitato automaticamente

---

### **Settare wp:**

Comandi

**watch** locazione

wp quando il programma scrive su locazione

**rwatch** locazione

wp quando il programma legge da locazione

**awatch** locazione

wp quando il programma scrive o legge locazione

### **Visualizzare wp:**

**info** watchpoints

equivale a "info breakpoints"

---

### **Condizioni per i bp/wp:**

Osservazioni:

- per i wp la condizione puo' essere settata solo con il comando "condition"
- devono essere condizioni booleane
- viene fermato il programma solo se la condizione indicata e' valutata vera
- e' possibile negare la condizione antepoendo il simbolo '!' alla condizione (! not in C)

### **Modificare/Aggiungere la condizione**

**condition** n condizione

aggiunge condizione al bp/wp numero n

### **Cancellare la condizione**

**condition** n (niente)

cancella la condizione al bp/wp numero n (in realta' setta la condizione a niente)

### **Eseguire lo stop del programma solo dopo m volte che si e' verificato il bp/wp**

**ignore** n m

setta il bp/wp numero n in modo che generi lo stop del programma solo dopo m volte che si e' verificato il bp/wp stesso

---

### **Eseguire comandi in automatico quando si verifica lo stop per un bp**

Comando **commands**

**commands** n

lista\_comandi

end

imposta la lista dei comandi da eseguire per il bp numero n

*Suggerimento:* mettere come primo comando "silent" per sopprimere il normale output del bp

es. Stampare il valore della variabile x, settando il bp sulla funzione prova, solo se x e' >0 e poi continuare l'esecuzione del programma. I comandi vengono indicati quando viene dichiarato il bp.

```
break prova if x>0
commands
silent
printf "x vale: %d\n",x
continue
end
```

---

### **Continuare l'esecuzione del programma**

Comando **continue** (abbreviato c o fg)

**continue** [n]

continua l'esecuzione del programma; se n e' indicato ignora per n stop l'ultimo bp su cui si e' fermato

Comando **step** (abbreviato s)

**step** [n]

continua l'esecuzione del programma fino alla prossima riga del codice sorgente; ATTENZIONE: se non si ha a disposizione il sorgente o si e' in una funzione senza informazioni di debug ritorna alla funzione chiamante.

Se n e' indicato esegue n step

Comando **stepi** (abbreviato si)

**stept** [n]

continua l'esecuzione del programma eseguendo ogni singola istruzione macchina  
Se n e' indicato esegue n stept

*Suggerimento:* dare il comando "display/i \$pc" per visualizzare in modo automatico l'istruzione che deve essere eseguita (vedere Visualizzazione automatica)

Comando **next** (abbreviato n)

**next** [n]

continua l'esecuzione fino alla prossima istruzione nello stack frame corrente; se l'istruzione e' una funzione, non entra nella funzione ma la esegue come fosse una singola istruzione  
Se n e' indicato esegue n next

Comando **nexti** (abbreviato ni)

**nexti** [n]

continua l'esecuzione fino alla prossima istruzione nello stack frame corrente; se l'istruzione e' una funzione, entra nella funzione  
Se n e' presente esegue n nexti

Comando **finish**

**finish**

continua l'esecuzione del programma fino alla terminazione della funzione corrente e stampa il valore di ritorno (se c'e')

Comando **until** (abbreviato u)

**until**

continua l'esecuzione del programma fino a quando le istruzioni eseguite hanno indirizzo minore dell'istruzione sulla quale si da il comando until.  
Utile nei cicli con salto all'indietro

**until** indirizzo

continua l'esecuzione del programma fino a quando viene raggiunta l'istruzione all'indirizzo "indirizzo"

---

**Vedere il disassemblato**

Comando **disassemble**

**disassemble**

visualizza il sorgente assembly della funzione in cui si trova l'istruzione corrente

**disassemble** indirizzo

visualizza il sorgente assembly della funzione in cui si trova l'istruzione a indirizzo

**disassemble** ind\_i ind\_f

visualizza il sorgente assembly delle istruzioni dall'indirizzo ind\_i (compreso) all'indirizzo ind\_f (escluso)

---

## **Alterare l'esecuzione del programma:**

### **Assegnare il valore ad una variabile**

Comando **set**

**set** nome\_var=valore  
setta la variabile nome\_var a valore

*Suggerimento:* se il nome di nome\_var coincide con un sottocomando di set usare la sintassi "set var nome\_var=valore"

Se si vuole effettuare il casting di una locazione di memoria

**set** {tipo\_dato}indirizzo=valore

es. Copiare 4 come intero nella locazione 0x83040

**set** {int}0x83040=4

### **Continuare ad un indirizzo diverso**

Comando **jump**

**jump** num\_linea  
riprende l'esecuzione dalla linea num\_linea

**jump** \*indirizzo  
riprende l'esecuzione dall'istruzione all'indirizzo indirizzo

Comando **set**

**set** \$pc=indirizzo  
la prossima istruzione sara' l'istruzione all'indirizzo indirizzo, ma a differenza di jump non riprende l'esecuzione

### **Ritornare da una funzione**

**return** [val\_ritorno]  
cancella l'esecuzione della funzione; se presente ritorna val\_ritorno coma valore di ritorno della funzione

### **Chiamare funzioni**

**call** nome\_funzione  
chiama la funzione nome\_funzione

---

## Esaminare i dati:

### Stampare il valore di una espressione

Comando **print** (abbreviato p)

**print**[/f] espressione

visualizza il valore di espressione; se presente /f utilizza il formato indicato

#### Formati di visualizzazione

x	intero esadecimale
d	intero decimale
u	intero decimale senza segno
o	intero ottale
t	intero binario
a	indirizzo
c	carattere
f	virgola mobile

es. Stampare il valore di \$pc in esadecimale

print/x \$pc

### Visualizzare il contenuto delle locazioni di memoria

Comando **x**

**x**[/nfu] indirizzo

visualizza il contenuto della locazione di memoria a indirizzo; se presenti utilizza i parametri /nfu

#### Parametri

n	intero decimale; indica quante unita' di memoria visualizzare partendo da in
f	formato di visualizzazione
s	stringhe null-terminated
i	istruzioni macchina
u	dimensione di una unita'
	b byte
	h mezza word
	w word
	g doppia word

es. Visualizzazione di 3 istruzioni da quella a indirizzo

x/3i indirizzo

*Suggerimento:* per visualizzare m unita' successive mantenendo lo stesso formato e' sufficiente dare il comando x/m

es. visualizzare 7 istruzioni macchina dopo aver dato il comando "x/3i"

x/7

---

## Visualizzazione automatica

### Settare una visualizzazione automatica

Comando **display**

**display**[/f] espressione  
visualizza il valore di espressione; se /f e' presente usa il formato indicato

Formati di visualizzazione

x	intero esadecimale
d	intero decimale
u	intero decimale senza segno
o	intero ottale
t	intero binario
a	indirizzo
c	carattere
f	virgola mobile
i	istruzione
s	stringa null-terminated

Se il formato e' i o s

**display**[/f] indirizzo  
visualizza il contenuto della locazionea indirizzo

### Visualizzare le visualizzazioni automatiche settate

**info display**  
visualizza tutte le le visualizzazioni automatiche settate

### Disabilitare le visualizzazioni automatiche settate

**disable display n**  
disabilita la visualizzazione automatica numero n

### Riabilitare le visualizzazioni automatiche settate

**enable display n**  
abilita la visualizzazione automatica numero n

### Cancellare le visualizzazioni automatiche settate

**delete display n**  
cancella la visualizzazione automatica numero n

---



## **Vedere tutti i registri e i loro valori**

**info** registers

visualizza tutti i registri e i loro valori, esclusi i registri floating point

**info** all-registers

visualizza tutti i registri e i loro valori, compresi i registri floating point

---

## **Esaminare la symbol-table**

**info** address simbolo

visualizza dove e' memorizzato simbolo

**whatis** espressione

visualizza il tipo di dato di espressione

**pptype** nome\_tipo

visualizza una descrizione del tipo di dato

**pptype** espressione

visualizza una descrizione di espressione; piu' completo di whatis

**info** source

visualizza il nome del file sorgente attuale

**info** sources

visualizza tutti i file sorgenti con informazioni di debug

**info** functions [expr]

visualizza nome e tipo di dati di tutte le funzioni; se e' presente expr, solo di quelle il cui nome contiene expr

**info** variables [expr]

visualizza informazioni su tutte le varibili; se e' presente expr, solo di quelle il cui nome contiene expr