#### Università di Pisa Corso di Laurea in Ingegneria Informatica

Guida al Debugging di programmi C, C++ e Assembler utilizzando il Data Display Debugger (DDD)

> a cura di **Marco Cococcioni**

## Cos'è DDD

- DDD sta per Data Display Debugger
- DDD è una intefaccia grafica (GUI) per lo GNU debugger, un programma di debug disponibile alla riga di comando.
- Come si avvia DDD alla diga di comando?
- Passo 1: compilare e linkare inserendo le informazioni per il debugger (opzione -g):

\$ g++ -g esl.cpp -o esl.exe

- Passo 2: avviare ddd:
  - \$ ddd es1.exe &

# Come appare DDD

<u>File Edit View Program Commands Status Source Data</u>	Help
(): main 🛛 🕲 🛱 👓 😳 ? 🍂 🎊 🏠 👘 OO 🤅 🥵 🔥 🕺	Undisp
Run <mark>Interrupt</mark> Step Stepi Next Nexti Until Finish Cont Kill Up Down Undo Redo Edit Make	
<pre>#include <iostream> using namespace std;</iostream></pre>	⊉.
int main() {	
int factorial = 1; int number = 0;	
int startnumber = 0;	
cout << "Please enter a number to calculate the factorial for: ";	
startnumber = number;	
while(number > 1)	_
{ factorial = factorial*number;	K
3	
cout << "The factorial of " << startnumber << " is " << factorial;	
Velcome to DDD 3.3.8 "Cinnamon" (sparc-sun-solaris2.9)	

### Customizzazione di DDD

- Numero di riga: Source  $\rightarrow$  Display Line Numbers
- Ancoraggio della barra degli strumenti: Edit → Preferences

	D:/cs/grad/jakubiak/Jakubiak/DebugLab/debuglab/sort.	-
<u>File</u> <u>E</u> dit	<u>View Program Commands Status Source Data</u>	<u>H</u> elp
(): main	🗸 说 🛱 💷 😳 🏹 🔬 🏷 Kotate Šet	Undisp
Run Inten		
1 #	- DDD Preferences	[ ]
2 # 3 # 4 u	General Source Data Startup Fonts Helpers	
5	Show Position and Breakpoints 🛛 🍈 as Glyphs 🔵 as Text Characters	
7 F 8	Tool Buttons Location 🔅 Command Tool 💿 Source Window	
9 i 10 {	Refer to Program Sources 🔿 by Path Name 🔘 by Base Name	
11 12	Find 🛛 Words Only 🗹 Case Sensitive	
13 14	Cache 🛛 🗹 Source Files 🗹 Machine Code	
15 16	Display Source Line Numbers	
17 18		
19 20	Tab Wirth Source Indentation Machine Code Indentation	
21 22		
23		
Copyri - Convri		
Copyri (qdb)	OK Reset Help	
	ons are located in command (000a).	

#### Come impostare i punti di interruzione

- Prima di mettere in esecuzione il programma occorre impostare uno o più **punti di interruzione** (*breakpoints*), altrimenti una volta avviato andrebbe immediatamente alla fine
- Per impostare un breakpoint:
  - click-pulsante-destro alla linea in cui si desidera impostare l'interruzione
  - Selezione dell'opzione Set Breakpoint
  - (ora dovrebbe comparire un segnale di stop all'inizio di quella linea)

#### Avvio del programma da debuggare

- In cima allo schermo, selezionare il bottone "Run" per avviare il programma
  - Run può anche essere trovato nel menu 'program'
- Il programma si arresterà in corrispondenza della riga in cui è posizionato il brackpoint
  - (La linea nera indica la prossima riga che verrà eseguita)

## Next, Hover, e Step

- Usare *Next* per muoversi nel programma una riga alla volta
- Dopo ogni Next è possibile posizionare il cursore sopra una variabile per vederne il valore
  - Questa operazione è chiamata hovering
- Step può essere usata in alternativa a Next
  - Step fa andare il programma una riga avanti, ma nel caso di chiamata di funzioni ci fa eseguire passo passo anche la funzione
  - Next fa andare il programma una riga avanti. Nel caso di chiamata di funzione, salta la chiamata e ci porta alla riga successiva alla chiamata stessa.

#### Correzione del programma (*Bug Fixing*)

- Per correggere il programma, una volta rilevato l'errore (baco):
  - chiudere DDD
  - aprire il file sorgente usando l'editor (gedit es1.cpp)
  - correggere il programma
  - ricompilare e rilinkare con opzione -g
  - riaprire l'eseguibile con DDD



## Inspezione delle variabili (1/2)

- Fare click sul pulsante destro sopra una variabile e poi scegliere Display nomeVariabile
  - questo provoca l'apertura del display editor
- Fare ancora click sul pulsante destro sopra al nome della variable e scegliere

#### **Display \*nomeVariabile**

(notare l'asterisco)

— DDD: /cs/grad/jakubiak/Jakubiak/DebugLab/debuglab/ -	J
File Edit View Program Commands Status Source Data Hel	,
(): youngest V S H Break Watch Print Display Plot Hide Rotate Set Undis	, , , ,
Run Interrupt Step Stepi Next Nexti Until Finish Cont Kill Up Down Undo Redo Edit Make	
1. friends	
name = 66 'B' $name = 77 'M'$ $name = 80 'P'$ $name = 90 'Z'$ $r$ $age = 60$ $age = 70$ $age = 100$ $age = 50$ $r$	
2: youngest (Eriend *) 0x10a1c	
<pre>11 Friend tmp; 12 Friend *list, *N, *youngest; 13 int nFriends, i; 14 nFriends = MakeFriends(friends); 15 cout &lt;&lt; "You have " &lt;&lt; nFriends &lt;&lt; " friends:" &lt;&lt; endl; 16 for (i = 0; i &lt; nFriends; i++) { 17 cout &lt;&lt; friends[i].name &lt;&lt;", age " &lt;&lt; friends[i].age &lt;&lt; endl; 18 } 19 youngešt = FindYoungest(friends, nFriends); 20 cout &lt;&lt; "Your youngest friend is: " &lt;&lt; endl; 21 cout &lt;&lt; youngest-&gt;name &lt;&lt;", age " &lt;&lt; youngest-&gt;age &lt;&lt; endl; 22 cout &lt;&lt; "Your oldest friend is:" &lt;&lt; endl; 23 youngest = FindOldest(friends, nFriends); 24 cout &lt;&lt; youngest-&gt;name &lt;&lt;", age " &lt;&lt; youngest-&gt;age &lt;&lt; endl; 25 return (0); 26 }</pre>	
Breakpoint 1, main () at sort.cpp:19 (gdb) graph display friends (gdb) graph display youngest (gdb) i	
△ Display 2: youngest (enabled, scope main, address 0×ffbff35c)	7

# Inspezione delle variabili (2/2)

- Le variabili possono essere visualizzate nell'area di display in diverse rappresentazioni:
  - /t nomeVar : binario
  - /d nomeVar : decimale
  - /h nomeVar : esadecimale
  - /o nomeVar : ottale
- Nel caso di programmi assembler si può visualizzare nella finestra di display (quella in alto) il contenuto dei registri come fossero variabili:
  - /t \$eax: mostra eax in bin
  - /d \$ebx: mostra ebx in decimale, ecc...
- Il contenuto dei registri può essere visualizzato anche nella finestra console di dello GNU debugger (GDB), nel seguente modo:
  - **i r eax** (che sta per info register eax)
- NB1: qui non serve \$, né %, prima del nome del registro
- NB2: esiste una limitazione: il contenuto dei registri a 8 e 16 bit (al, ah, ax, bl, ...) non può essere visualizzato

#### Per tornare indietro nell'esecuzione

- Per andare indietro rispetto alla linea di interruzione
  - clickare e trascinare la freccia in alto, alla linea desiderata

-	DD	D:	/cs	5/g	j ra	ια/	Ja	IKU		- Curv	, J.						. g.		· ·			<u> </u>	· ·				
File	Edi	t <u>v</u>	'iew	Pn	ogra	m	Co	mma	ands	S	tatus	s <u>s</u>	Sourc	e <u>C</u>	ata												Help
0:	sort	.ср	p:20	Ĭ			V	Loc	9 okup	) Fit		Bre	ak	GG Vatc	▼ i P	<b>?</b>	Dis	DIRV	N. PI	∧_ ot	G Sh	ow ow	Rot	▶ ate	New Sector	de la t	) Undis
Run	Intern	upt	Step	Ste	pi N	Vext	Ne	exti	Unti	I Fi	nish	Cor	it K	ill Up	Do	/vn	Undo	Re	do	Edit	Ma	ake					
	• •																										
	• •		• •	•		:	:			:	· ·		:	· ·			· ·		•	:	:	•	· ·	:			· ·
	• •	•				•	•					•		• •		·										•	
1	2 3	Fri int	iend : nF	*] rie de	lis end	t, s, Mak	*N i;	, " :ri:	*yoi	ung c(f	est ric	t;	.).														1
	234587890122345	Fri int cou for you cou cou cou you cou	iend inF ien it < iout inge it < inge it < iurn	*  rie ds < ' < ' < ' st < ' ((	lis end: "Yo "Yo ; You "You "You ));	t, s, Mak u h rie Fin ur Fin nge	*N i; eF av od yo st ol st	irie e nFr ls[i des ides ides ides	*yon = nd: < i i].n nge: ges: nam = st nam	ung < n nds nam st( fri (fr e <	iest Fri Fri is; i fri fri enc ier ier	t; iends i++) iend , ag nds, , ag	5); ls is; je ; nl ; nl je	<< " ge " " " << ? Frie " <<	fr << yo end nds yo	ien fr ds) ung l; ung	ds: ien ; est est	" < .ds[ .−>a .−>a	(< ( (i]) (ge	end . ag <<	l; e < er	< ( ndl;	end ; ;	1;			
1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	2 3 4 5 7 8 9 20 21 22 3 4 5 8 9 20 21 22 3 4 25 8 9 20 21 22 3 4 5 7 8 9 20 21 22 3 4 5 7 8 9 20 21 22 3 24 5 5 8 9 20 21 22 3 24 5 5 8 9 20 21 21 22 3 24 5 5 8 9 20 21 21 21 21 21 21 21 21 21 21 21 21 21	Fri int nFr cou for you cou cou you cou ret	iend rien rien (it cout unge ut < unge t < t 1,	*1 rie ds < ' < st < } < (0 ma	lis end: "You 0; < f = "You You "You ));	t, s, Mak i rie Fin ur Fin nge ()	*N i; eF iav id yo st id of st	irie e nFi s[s[i des des ldes i des t s	*yo end: " < rie i].u nge: namu st ` sor'	ung s(f nds nam st( fri (fr e < t.c	iest Frie Fri frie enc ier :<",	t; iends i++) iend , ag d is , ag :19	s); ls { is; ge ' ; nl ge '	<< " ge " nFr ' << < Frie	fr << yo end yo	ien fr ds) end ung l; );	ds: ien ; est	" < ds[ >a >a	(< ( (i]) uge	end . ag <<	l; e < er	<< ( ndl;	end ; ;	1;			

### Materiale addizionale

# **Conditional Breakpoints**

- What if you had 10,000 friends? It would take a long to check each assignment.
- Instead, first make sure that the loop reaches the youngest friend.
- Use a conditional breakpoint
  - Insert a breakpoint
  - Right-click the break point and choose properties
  - Enter the condition (in C code) when, if true, the code should pause.

-	DDD	: /cs	s/grad	1∕ja	kub	iak/J	aku	biak	/De	bug	зLа	ıb/	de	buç	gla	ıb/	frie	en		
File	Edit	⊻iew	Program	Cor	nmand	s Stati	is <u>S</u> i	ource	Data										H	elp
0:	friend	ls.cpp	p:20į́	V	ම Looku	Find»	Clea	r Vat	h Pr	int I	Jispia Algait	K ' K	V/ Plot	€ Sho	w.	() Rotal	te.	1901 H	Un	Ø, Jisp
Run	Interrupt	Step	Stepi Ne	xt Ne	xti Ur	til Finisł	Cont	Kill U	Dow	n Un	ndo   F	Redo	Edit	Ma	ke					
:	· ·		· · ·	· ·		· ·	· ·	· ·	:	· ·	:	· ·	:			:	•			:
				· ·	· ·			· ·												·
•				• •	• •	• •		• •			•		•		•	•	•		•	· .
	15 F 16 i 17 f 18 19 21 22 }	nt i; or (i if ( }	= 0; !young younge	yest i <= est st =	= - n1 11 &1	Breakp Conditio	DD oint 5 on	D:P	ope •	ertie Die o	2 <b>5 :</b> Jisabi age	Bro	eak Misol	po tele	int t	t 5				
	23 r 24 } 25 //	eturn	(young	est),	;	Ignore	Count	I							_					I
	26 //-		icurn a	pun		Comma	ands	R	cord		End		Edi	t >>				⊫		
Fi (g Br (g	ndYoun db) br eakpoi db) ĭ	gest eak f nt 5	(friends riends at Ox1	ds=0: .cpp OceO	×21 – :2( : 1	Appl	у			Clos	e				H	Help				
				_																

## Watchpoints

- Often you have a program and, at some point, an important value gets set incorrectly
  - e.g., using (x = 0) instead of (x == 0)
  - Suppose you know that
    - A line 6, x is set to 7
    - At line 9,234 x = 0 (but x should still be 7).
  - How can you quickly find the line where x was incorrectly set.
    - A watchpoint allows you to watch for a variable to change