le display de DDD

- récuperer le fichier test.c
- compiler avec l'option -g (dans le cas contraire un message serait envoyer : /elf/srart.S non ...)
 - 1. gcc -g -o test test.c
- lancer DDD
 - 1. ddd ./test

a bbb. //ome/dupont/bbb/test.c			کا لکارکا
<u>File Edit View Program Commands Status Source Data</u>			<u>H</u> elp
(): test.c:153	no» Clear Vaton Print	Lastay Mr. C.	Sotate Set Units
		Run Interrupt Step Stepi Next Nexti	
149 /************************************		Cont Kill	
<pre>150 int main() { 151 liste_t L; 152 L=NULL; 153 L = insereTete(1,insereTete(2,insereTete(4,insereTete(5,NULL)))); 154 printListe(L); 155 L = insereQueue(7,(insereQueue(6,L))); 156 printListe(L); 157 // 157 //</pre>		Up Down Undo Redo Edit Make	
158 159 printListe(L);			
<pre>160 161 L= inserePosition(L,0,0); 162 printListe(L); 163 L= inserePosition(L,3,3); 164 printListe(L); 165 L= inserePosition(L,1,1);</pre>			
Dump of assembler code from 0x80486d4 to 0x80487d4: 0x080486d4 <main+28>: movl \$0x0,0xfffffffc(%ebp) ● 0x080486db <main+35>: movl \$0x0,0x4(%esp) 0x080486ea <main+43>: movl \$0x5,(%esp) 0x080486ea <main+5>: call 0x8048430 <inseretete> 0x080486ef <main+5>: mov %eax,0x4(%esp) 0x080486f3 <main+5>: call 0x8048430 <inseretete> 0x080486f3 <main+6>: call 0x8048430 <inseretete> 0x080486ff <main+71>: mov %eax,0x4(%esp) 0x080486ff <main+71>: mov %eax,0x4(%esp) 0x080486ff <main+71>: mov %eax,0x4(%esp) 0x080486ff <main+71>: mov %eax,0x4(%esp) 0x08048703 <main+75>: movl \$0x2,(%esp)</main+75></main+71></main+71></main+71></main+71></inseretete></main+6></inseretete></main+5></main+5></inseretete></main+5></main+43></main+35></main+28>			
(gdb) break test.c:153 Breakpoint 1 at 0x80486db: file test.c, line 153. (gdb)			
Δ Breakpoint 1 at 0x80486db; file test.c, line 153.			Î.
🔣 💽 🎲 sort.c - Kate LDDD: /home/dupont/DDD:	2 A X	13:03 2007-02-14	ļ 🗎 🚺
2 placer un stop en 154			

- 2. placer un stop en 154.
- (gdb) run Brealpoint 1,main () at test.c:154
- 3. cliquez sur Run
- 4. puis sur next
- (gdb) next
- 5. cliquez sur L
- 6. L'apparaît dans la ligne de saisie du haut
- 7. En plaçant quelques instants le curseur sur L des informations apparaissent : (liste_t) 0x804a038*



Il s'agit maintenant de VISUALISER la liste.

- 1. Cliquez sur le bouton Display : Display visualise ce qui est ecrit dans la ligne de saisie.
- (gdb) graph display L

🗧 DDD: /home/dupont/DDD/test.c						
<u>File Edit View Program Commands Status Source Data</u>				Help		
(): U	00 ? Uatch Print Dis	play Plot H	🔾 🗸 🔂	e Set Undis		
$\frac{1: L}{(1:1:t_{e}, t) 0:804a038} = \frac{1}{next} = \frac{1}{0.004a028}$						
<pre>149 /************************************</pre>	Run Interrupt Step Step			· · · · · · · · · · · · · · · · · · ·		
0x08048713 <main+91>: movl \$0x1,(%esp) 0x0804871a <main+98>: call 0x8048430 <inseretete> 0x0804871f <main+103>: mov %eax,0xffffffc(%ebp) 0x08048722 <main+106>: mov 0xffffffc(%ebp),%eax 0x08048725 <main+109>: mov %eax,(%esp) 0x08048725 <main+112>: call 0x8048672 <printliste> 0x08048726 <main+117>: mov 0x4fffffc(%ebp),%eax 0x08048726 <main+117>: mov %eax,0xffffffc(%ebp),%eax 0x08048730 <main+12>: call 0x8048672 <printliste> 0x08048734 <main+12>: mov %eax,0x4(%esp) 0x0804873b <main+131>: call 0x804867 <inserequeue></inserequeue></main+131></main+12></printliste></main+12></main+117></main+117></printliste></main+112></main+109></main+106></main+103></inseretete></main+98></main+91>	Next Next Until Finis Cont Kill Up Down Undo Redd Edit Make	i h 1 2				
Breakpoint 1, main () at test.c:154 (gdb) next (gdb) graph display L (gdb) Quit (gdb) graph display &L (gdb) graph undisplay &L (gdb) set variable L = (liste_t) 0x804a038 (gdb) graph display *L dependent on 1 (gdb)						
△ Display 3: *L (enabled, scope main, address 0x804a038) Image: State of the state	₹ 1 1 1 1 1 1 1 1 1 1	13:53 007-02-14				

L est un pointeur sur une liste dont la valeur est = 0x804a038.

- 1. tapez dans la ligne de saisie *L
- 2. cliquez sur Display

Vous pouvez cliquez suur le contenue de *L

- 1. cliquez par exemple sur next=0x804a028
- remarquez L->next dans la barre de saisie
- 2. Cliquez sur Val=1
- remarquer L->val dans la barre de saisie

Faire apparaître la suite de la liste

- 1. cliquez sur next
- 2. cliquez sur display
- L->next->next apparaît dans la barre de saisie
- et ensuite de suite.

کے DDD: /home/dupont/DDD/test.c		(- 🗗 🗙
File Edit View Program Commands Status Source Data			Help
(): L->next->next->next->next	ak Vatch Print Disp*	MA Q Ø	e Set Undisp
Image: Indext / Hext	ak uato, print 6554		
1/2 ///////////////////////////////////			
<pre>150 int main() { 150 int main() { 151 liste_t L; 152 L=NULL; 152 l=NULL; 153 L = insereTete(1,insereTete(2,insereTete(4,insereTete(5,NULL)))); 154 printListe(L); 155 L = insereQueue(7,(insereQueue(6,L))); 156 printListe(L); 157 // 158 159 printListe(L); 160 0x08048713 <main+91>: mov1 \$0x1,(%esp) 0x08048714 <main+91>: mov1 \$</main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></main+91></pre>	Run Interrupt Step Stepi Next Nexti Until Finish Cont Kill		г ————————————————————————————————————
 0x08048722 <main+1052: 0xffffffc(%ebp),%eax<="" li="" mov=""> 0x08048725 <main+1092: %eax,(%esp)<="" li="" mov=""> 0x08048728 <main+112>: call 0x8048672 <printliste></printliste></main+112> 0x08048724 <main+117>: mov 0xffffffc(%ebp),%eax</main+117> 0x08048730 <main+12>: mov %eax,0x4(%esp)</main+12> 0x08048734 <main+12>: movl \$0x6,(%esp)</main+12> 0x08048735 <main+131>: call 0x804867 <inserequeue></inserequeue></main+131> </main+1092:></main+1052:>	Up Down Undo Redo Edit Make		
<pre>(gdb) graph undisplay 2 (gdb) set variable L = (liste_t) 0x804a038 (gdb) graph display *L dependent on 1 (gdb) graph display *(L->next) dependent on 3 (gdb) graph display *(L->next->next) dependent on 5 (gdb) graph display *(L->next->next) dependent on 5 (gdb) graph display *(L->next->next->next) dependent on 6 IDisabling display 7 to avoid infinite recursion. (gdb) graph display L->next->next->next (gdb)</pre>			
△ Display 8: L->next->next->next (enabled, scope main, address 0x804a00c)			- F
Image: Second	² ∛ 14:0 4 № 2007-0)3	