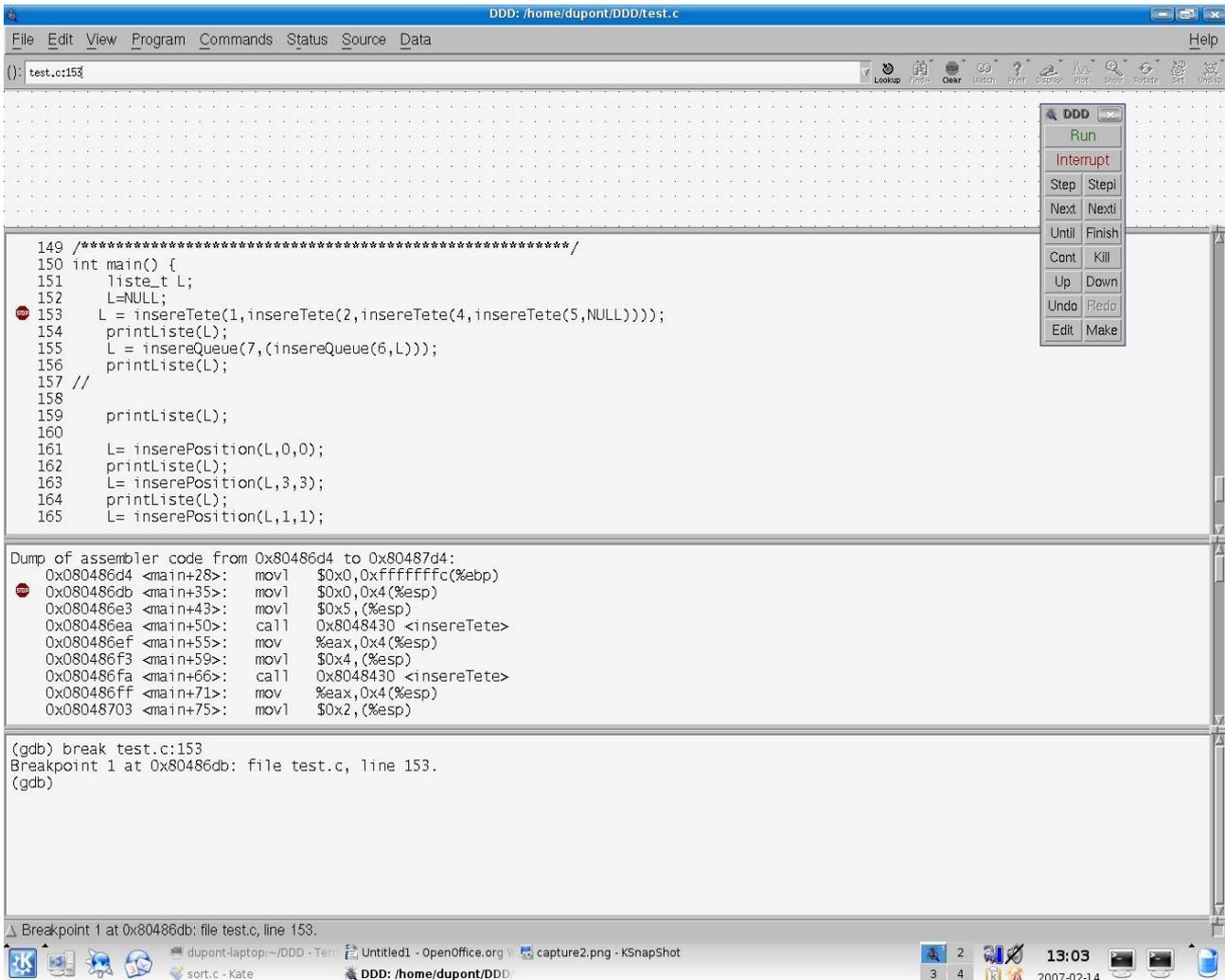
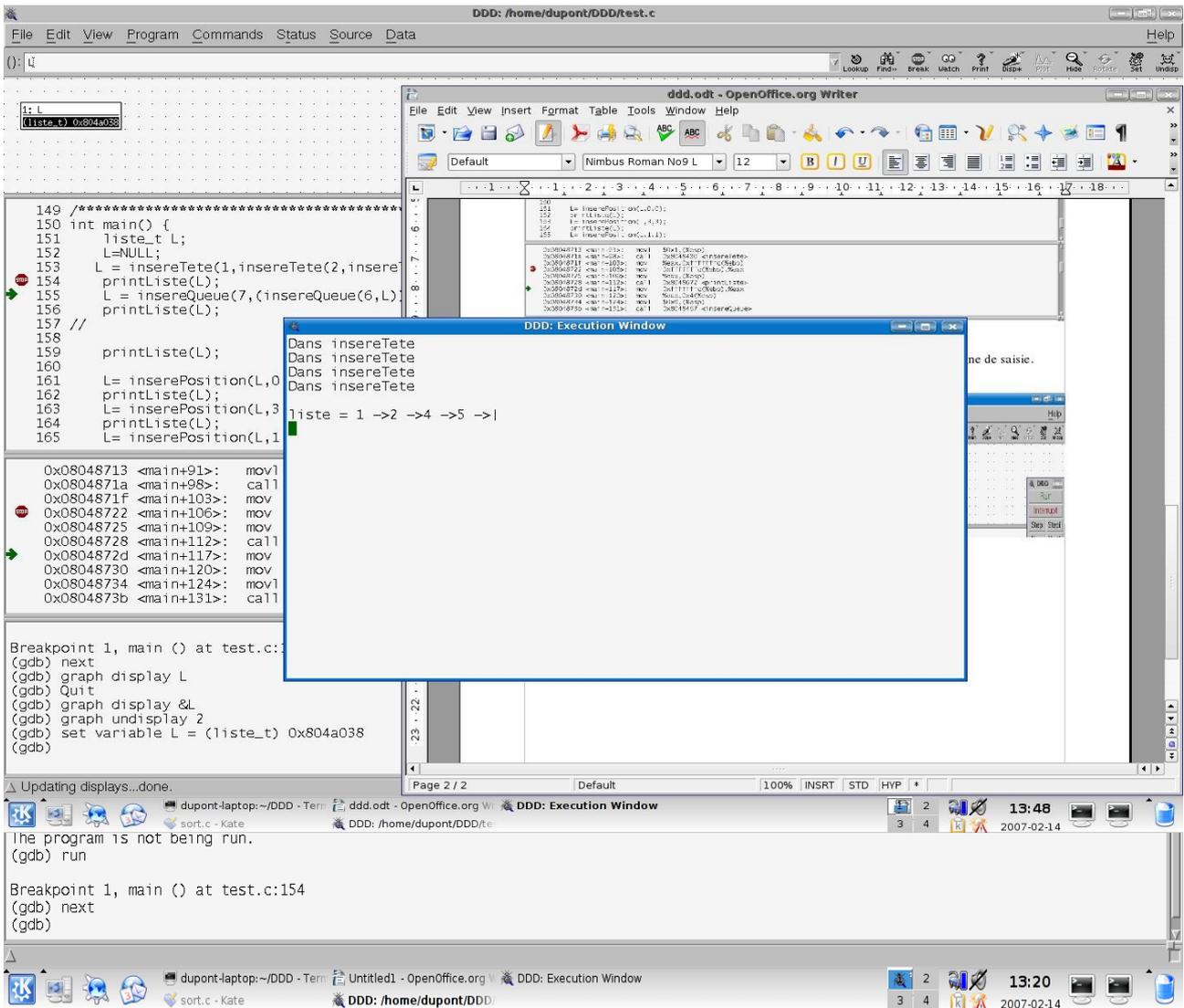


# le display de DDD

- récupérer 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

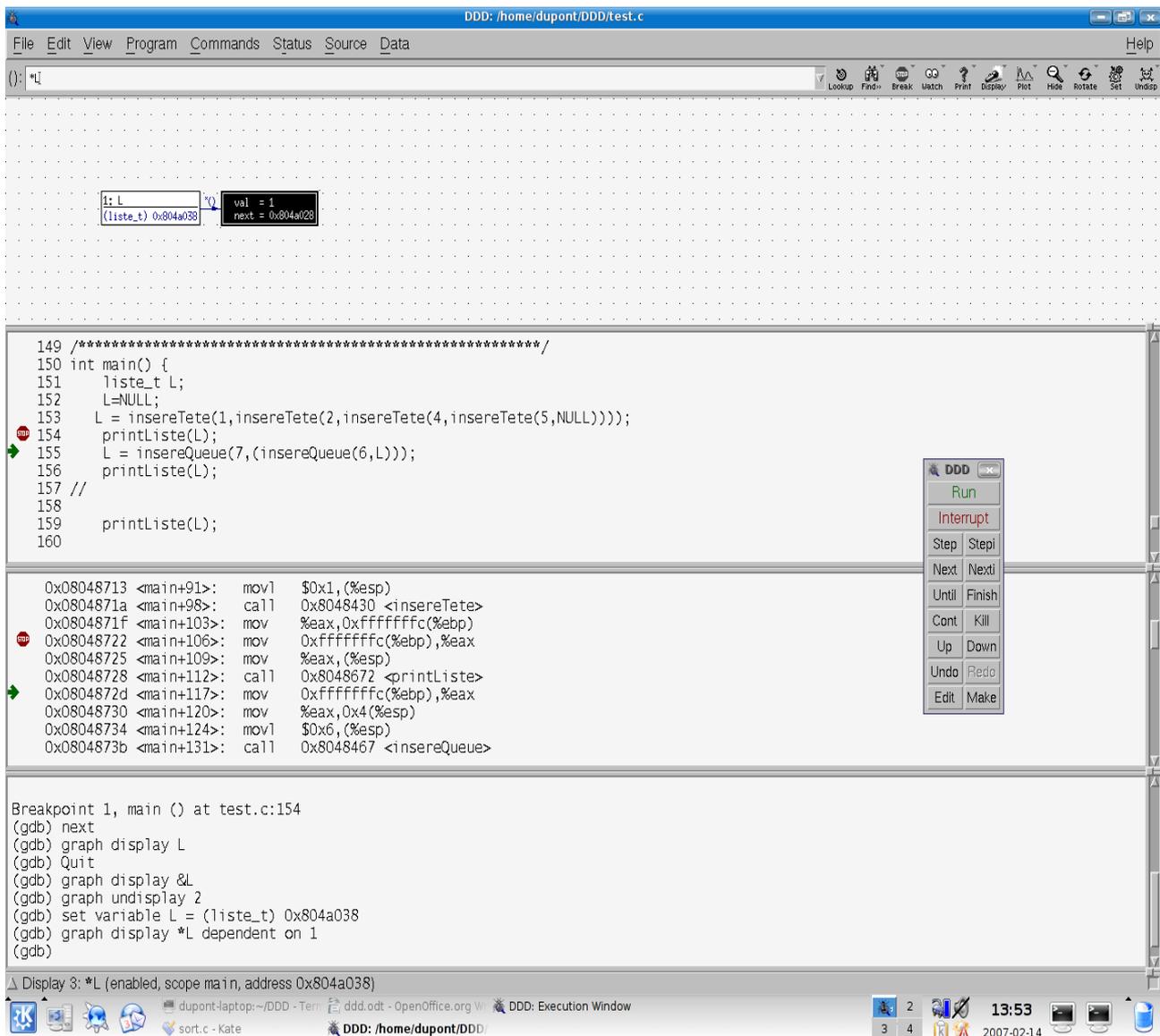


2. placer un stop en 154.
  - (gdb) run Brealpoint 1,main () at test.c:154
3. cliquez sur Run
4. puis sur next
  - (gdb) next
5. cliquez sur L
6. L apparaît dans la ligne de saisie du haut
7. En plaçant quelques instants le curseur sur L des informations apparaissent : (liste\_t) 0x804a038\*



Il s'agit maintenant de VISUALISER la liste.

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



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 cliquer sur le contenu 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.

DDD: /home/dupont/DDD/test.c

File Edit View Program Commands Status Source Data Help

() L->next->next->next->next

8: L->next->...xt->next  
(struct liste\_s \*) 0x0

```

1: L
(liste_t) 0x804a038
    val = 1
    next = 0x804a028
    next
    val = 2
    next = 0x804a018
    next
    val = 4
    next = 0x804a008
    next
    val = 5
    next = 0x0
  
```

```

149 /*****/
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 //
158 //
159     printListe(L);
160
0x08048713 <main+91>: movl    $0x1, (%esp)
0x0804871a <main+98>: call   0x8048430 <insereTete>
0x0804871f <main+103>: mov    %eax, 0xffffffff(%ebp)
0x08048722 <main+106>: mov    0xffffffff(%ebp), %eax
0x08048725 <main+109>: mov    %eax, (%esp)
0x08048728 <main+112>: call  0x8048672 <printListe>
0x0804872d <main+117>: mov    0xffffffff(%ebp), %eax
0x08048730 <main+120>: mov    %eax, 0x4(%esp)
0x08048734 <main+124>: movl   $0x6, (%esp)
0x0804873b <main+131>: call  0x8048467 <insereQueue>
  
```

```

(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 4
(gdb) graph display *(L->next->next->next) dependent on 5
(gdb) graph display *(L->next->next->next->next) dependent on 6
Disabling display 7 to avoid infinite recursion.
(gdb) graph display L->next->next->next->next
(gdb)
  
```

Display 8: L->next->next->next->next (enabled, scope main, address 0x804a00c)

dupont-laptop:~/DDD - Terminal ddd.oct - OpenOffice.org Writer DDD: Execution Window

sort.c - Kate DDD: /home/dupont/DDD/ 2 14:03 3 4 2007-02-14