



COMPTE-RENDU

TP numéro 4-5 [LO22]



11 MAI 2016

Marion Chan-Renous-Legoubin & Laviolette Etienne

Compte rendu TP4-5

Rappel des objectifs du TP 4:

L'objectif principal de ce TP numéro 4 de LO22 est de mettre en œuvre le processus de test d'un programme à travers l'outil GCC. Nous allons donc nous intéresser à l'exécution de plusieurs programmes et à la mesure de couverture. Pour chaque programme, nous allons préparer une liste des tests à exécuter pour couvrir chaque instruction de celui-ci, instrumenter le programme, puis exécuter l'ensemble des tests en mesurant la couverture à chaque fois.

I. PROGRAMME 1

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int i, j, total;
6      printf("Entrer une valeur pour j : \n");
7      scanf("%d",&j) ;
8      total = 0;
9      for (i = 0; i < j; i++)
10         total += i;
11     if (total != 45)
12         printf ("Failure\n");
13     else
14         printf ("Success\n");
15     return 0;
16 }
17
```

- TESTS A EFFECTUER

Nous savons que $0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 = 45$.

Ainsi, afin de tester chaque instruction du programme, il est nécessaire d'effectuer deux tests:

- Un test avec une valeur de j différente de 10 pour tester l'instruction de la ligne 12
- Un test avec une valeur de j égale à 10 pour tester l'instruction de la ligne 14

- EXECUTION DES TESTS

- Test avec une valeur de j différente de 10

Nous avons testé le programme avec j ayant pour valeur 3.

Compte rendu TP4-5

```
-:      0:Source:programmel.c
-:      0:Graph:programmel.gcno
-:      0:Data:programmel.gcda
-:      0:Runs:1
-:      0:Programs:1
-:      1:#include <stdio.h>
-:      2:
1:      3:int main()
-:      4:{
-:      5: int i, j, total;
1:      6: printf("Entrer une valeur pour j : \n");
1:      7: scanf("%d",&j) ;
1:      8: total = 0;
4:      9: for (i = 0; i < j; i++)
3:     10:     total += i;
1:     11: if (total != 45)
1:     12:     printf ("Failure\n");
-:     13: else
####:    14:     printf ("Success\n");
1:     15: return 0;
-:     16:}
```

La sortie de ce test est : Failure

ANALYSE DU TEST :

Nous pouvons voir que la boucle de l'instruction de la ligne 4 est bien appelée. Nous pouvons également voir que la ligne 12 est appelée, mais la ligne 14 ne l'est pas. Ainsi, ceci est cohérent avec le fait que la valeur de j est différente de 10, donc on rentre dans le "if" de la ligne 11.

- Test avec une valeur de j égale à 10

Compte rendu TP4-5

```
-: 0:Source:programmel.c
-: 0:Graph:programmel.gcno
-: 0:Data:programmel.gcda
-: 0:Runs:1
-: 0:Programs:1
-: 1:#include <stdio.h>
-: 2:
1: 3:int main()
-: 4:{
-: 5: int i, j, total;
1: 6: printf("Entrer une valeur pour j : \n");
1: 7: scanf("%d",&j) ;
1: 8: total = 0;
11: 9: for (i = 0; i < j; i++)
10: 10:     total += i;
1: 11: if (total != 45)
#####: 12:     printf ("Failure\n");
-: 13: else
1: 14:     printf ("Success\n");
1: 15: return 0;
-: 16:}
```

La sortie de ce test est : Success.

ANALYSE DU TEST :

Nous pouvons voir que la boucle de l'instruction de la ligne 4 est bien appelée. Nous pouvons également voir que cette fois ci la ligne 14 est appelée, mais la ligne 12 ne l'est pas. Ainsi, ceci est cohérent avec le fait que la valeur de j est égale à 10, donc on rentre dans le "else" de la ligne 13.

PROGRAMME 2

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int i, j, total;
6
7      scanf("%d",&j) ;
8      total = 0;
9
10     for (i = 0; i < j; i++)
11         total += i;
12     if (total == 45)
13         printf ("Success\n");
14
15     return 0;
16 }
```

- TESTS A EFFECTUER

Ce programme est similaire au précédent.

Ainsi, afin de tester chaque instruction du programme, il est nécessaire d'effectuer deux tests:

- Un test avec une valeur de j différente de 10 pour tester que la ligne 13 n'est pas appelé dans le cas où le total n'est pas égal à 45
- Un test avec une valeur de j égale à 10 pour tester l'instruction de la ligne 13

- EXECUTION DES TESTS

Compte rendu TP4-5

- Test avec une valeur de j différente de 10

Nous avons testé le programme avec j ayant pour valeur 5.

```
-:      0:Source:programme2.c
-:      0:Graph:programme2.gcno
-:      0:Data:programme2.gcda
-:      0:Runs:1
-:      0:Programs:1
-:      1:#include <stdio.h>
-:      2:
1:      3:int main()
-:      4:{
-:      5: int i, j, total;
-:      6:
1:      7: scanf("%d",&j) ;
1:      8: total = 0;
-:      9:
6:     10: for (i = 0; i < j; i++)
5:     11:     total += i;
1:     12: if (total == 45)
#####: 13:     printf ("Success\n");
-:     14:
1:     15: return 0;
-:     16:}
```

Compte rendu TP4-5

Sortie du test : rien n'est affiché à l'écran

ANALYSE DU TEST :

Nous pouvons voir que la boucle de l'instruction de la ligne 10 est bien appelée. Nous pouvons également voir que cette fois ci la ligne 13 n'est pas appelée. Ceci est cohérent avec le fait que la valeur de j n'est pas égale à 10, donc on ne rentre pas dans le "if" de la ligne 12. Il n'y a pas d'instruction à effectuer lorsqu'on ne rentre pas dans ce "if", ce qui explique pourquoi il ne s'affiche rien à l'écran.

- Test avec une valeur de j égale à 10

```
-:      0:Source:programme2.c
-:      0:Graph:programme2.gcno
-:      0:Data:programme2.gcda
-:      0:Runs:1
-:      0:Programs:1
-:      1:#include <stdio.h>
-:      2:
1:      3:int main()
-:      4:{
-:      5: int i, j, total;
-:      6:
1:      7: scanf("%d",&j) ;
1:      8: total = 0;
-:      9:
11:     10: for (i = 0; i < j; i++)
10:     11:         total += i;
1:     12: if (total == 45)
1:     13:         printf ("Success\n");
-:     14:
1:     15: return 0;
-:     16:}
```


Compte rendu TP4-5

test est : Success.

ANALYSE DU TEST :

Nous pouvons voir que la boucle de l'instruction de la ligne 10 est bien appelée. Nous pouvons également voir que cette fois ci la ligne 13 est appelée. Ceci est cohérent avec le fait que la valeur de j est égale à 10, donc on rentre dans le "if" de la ligne 12 et l'instruction de la ligne 13.



COMPTE-RENDU

TP numéro 4-5 [LO22]



11 MAI 2016

Marion Chan-



Renous-

Compte rendu TP4-5

Rappel des objectifs du TP 5 :

L'objectif principal de ce TP numéro 5 de LO22 est d'analyser plusieurs programmes en C à travers l'outil Splint, afin de les corriger. Pour chaque programme, nous allons donc le compiler avec l'outil GCC puis exécuter la commande adéquate afin de l'analyser avec Splint.

PROGRAMME 1

```
1  |#include <stdio.h>
2
3  |printarray (Anarray)
4
5  |int Anarray;
6  |{
7  |    printf ("%d",Anarray);
8  |}
9
10 |main()
11 |{
12 |    int Anarray[5]; int i; char c;
13 |    printarray (Anarray);
14 |    printarray (Anarray, c, i);
15 |}
16
```

ANALYSE DU PROGRAMME

A première vue, ce programme déclare tout d'abord (ligne3) le prototype d'une fonction. Ensuite lignes 5-8 nous avons les instructions de cette fonction. Puis nous avons le main de ce programme qui semble utiliser à deux reprises cette fonction (lignes 13 et 14), avec des paramètres différents.

ANALYSE INFORMATIQUE

Nous avons analysé ce programme avec l'outil Splint.

Compte rendu TP4-5

```

elaviole@smestaens18j:/voletu/userix/users/elaviole/Documents/L022-TP4_TP5/T05$ splint progr1.c
Splint 3.1.2 --- 04 Aug 2013

progr1.c: (in function printarray)
progr1.c:8:2: Path with no return in function declared to return int
  There is a path through a function declared to return a value on which there
  is no return statement. This means the execution may fall through without
  returning a meaningful result to the caller. (Use -noret to inhibit warning)
progr1.c: (in function main)
progr1.c:13:13: Function printarray expects arg 1 to be int gets int [5]:
      Anarray
  Types are incompatible. (Use -type to inhibit warning)
progr1.c:13:2: Return value (type int) ignored: printarray(Anarray)
  Result returned by function call is not used. If this is intended, can cast
  result to (void) to eliminate message. (Use -retvalint to inhibit warning)
progr1.c:14:13: Function printarray expects arg 1 to be int gets int [5]:
      Anarray
progr1.c:14:2: Function printarray called with 3 args, expects 1
progr1.c:14:22: Variable c used before definition
  An rvalue is used that may not be initialized to a value on some execution
  path. (Use -usedef to inhibit warning)
progr1.c:14:25: Variable i used before definition
progr1.c:14:2: Return value (type int) ignored: printarray(Anarr...
progr1.c:15:2: Path with no return in function declared to return int
progr1.c:3:1: Function exported but not used outside progr1: printarray
  A declaration is exported, but not used outside this module. Declaration can
  use static qualifier. (Use -exportlocal to inhibit warning)
progr1.c:5:4: Definition of printarray

Finished checking --- 10 code warnings

```

INTERPRETATION DES RESULTATS

Nous pouvons voir que Splint nous affiche 10 warnings. Nous pouvons voir qu'il nous indique que la fonction ne retourne aucune valeur, que les appels de la fonction ne sont pas corrects par rapport au prototype qui a été déclaré, que le main ne renvoie aucune valeur, et qu'un tableau ne peut être affiché si il n'a pas été initialisé préalablement.

CORRECTION PROPOSEES

Nous avons corrigé ce programme en créant deux fonctions "printarray", une ayant un paramètre, et l'autre en ayant trois, en ajoutant une valeur de retour(0) au main et en initialisant le tableau Anarray correctement.

Compte rendu TP4-5

```
1  #include <stdio.h>
2
3  static void printarray1(int Anarray[], char c, int i)
4
5  {
6      int j=0;
7      for (j=0;j<i;j++)
8          printf("%d %c", Anarray[j], c);
9  }
10 static void printarray(int Anarray[])
11
12 {
13     int j=0;
14     for (j=0;j<5;j++)
15         printf("%d", Anarray[j]);
16 }
17
18 int main()
19 {
20     int j;
21     int i=5;
22     char c='x';
23     int Anarray[5];
24     for (j=0;j<5;j++)
25         Anarray[j]= 5;
26
27     printarray(Anarray);
28     printarray1(Anarray, c, i);
29     return 0;
30 }
```

ANALYSE INFORMATIQUE DU PROGRAMME CORRIGE

```
elaviole@smestaens18j:/voletu/user1x/users/elaviole/Documents/L022-TP4_TP5/TD5$ splint prog1_corrected.c
Splint 3.1.2 --- 04 Aug 2013

Finished checking --- no warnings
```

L'analyse par l'outil Splint du programme corrigé ne présente plus de warning. Ainsi, nous pouvons voir que la correction proposée du programme a été efficace.

Compte rendu TP4-5

PROGRAMME 2

```
1  main()
2  {
3      int i = 4;
4      for (;i < 4; i++);
5      for (;i > 0; i++);
6  }
7  }
```

ANALYSE DU PROGRAMME

Nous pouvons voir que ce programme crée une variable *i* de type integer et l'initialise avec la valeur 4 (ligne 3) et deux boucles for (4 et 5).

ANALYSE INFORMATIQUE

Nous avons analysé ce programme avec l'outil Splint.

```
elaviole@smestaens18]:/voletu/userix/users/elaviole/Documents/L022-TP4_TP5/T05$ splint progr2.c
Splint 3.1.2 --- 04 Aug 2013

progr2.c: (in function main)
progr2.c:6:2: Path with no return in function declared to return int
  There is a path through a function declared to return a value on which there
  is no return statement. This means the execution may fall through without
  returning a meaningful result to the caller. (Use -noret to inhibit warning)

Finished checking --- 1 code warning
```

INTERPRETATION DES RESULTATS

Nous pouvons voir que l'analyse par Splint du code de ce programme nous affiche un warning. Il nous informe ainsi que le main ne retourne aucune valeur.

CORRECTION PROPOSEES

Nous corrigeons ce programme en ajoutant une valeur de retour(0) au main.

```
1  int main()
2  {
3      int i = 4;
4      for (;i < 4; i++);
5      for (;i > 0; i++);
6      return 0;
7  }
8  }
```

ANALYSE INFORMATIQUE DU PROGRAMME CORRIGE

```
elaviole@smestaens18]:/voletu/userix/users/elaviole/Documents/L022-TP4_TP5/T05$ splint prog2_corrected.c
Splint 3.1.2 --- 04 Aug 2013

Finished checking --- no warnings
```

Compte rendu TP4-5

L'analyse par l'outil Splint du programme corrigé ne présente plus de warning. Ainsi, nous pouvons voir que la correction proposée du programme a été efficace.

PROGRAMME 3

```

1  main()
2  {
3      int i = 4;
4      int a[7];
5      int j;
6
7      for (j= 0; j <=7; j++)
8          a[j] = j;
9      i = 5 / (i - 4);
10 }
11

```

ANALYSE DU PROGRAMME

Nous pouvons voir que ce programme crée une variable *i* de type integer et l'initialise avec la valeur 4, un tableau *a* de 7 entiers, et une variable *j* de type integer non initialisée. Nous pouvons également voir une boucle for (ligne 7), qui affecte à chaque case du tableau la valeur de l'indice de la case de celui-ci, et une instruction (ligne 9) qui affecte à la valeur *i* la valeur $5/(i-4)$.

ANALYSE INFORMATIQUE

Nous avons analysé ce programme avec l'outil Splint.

```

elaviole@mestaens18j:~/voletu/userix/users/elaviole/Documents/L022-TP4/TP5/TD5$ splint progr3.c
Splint 3.1.2 --- 04 Aug 2013

progr3.c: (in function main)
progr3.c:10:2: Path with no return in function declared to return int
  There is a path through a function declared to return a value on which there
  is no return statement. This means the execution may fall through without
  returning a meaningful result to the caller. (Use -noret to inhibit warning)

Finished checking --- 1 code warning

```

INTERPRETATION DES RESULTATS

Nous pouvons voir que l'outil Splint nous affiche un warning. Il nous informe ainsi que le *main* ne retourne aucune valeur.

CORRECTION PROPOSEES

Nous corrigeons ce programme en ajoutant une valeur de retour(0) au *main*.

Compte rendu TP4-5

```
1  int main()  
2  {  
3      int i = 4;  
4      int a[7];  
5      int j;  
6  
7      for (j= 0; j <=7; j++)  
8          a[j] = j;  
9      i = 5 / (i - 4);  
10     return 0;  
11 }
```

ANALYSE INFORMATIQUE DU PROGRAMME CORRIGE

```
elaviole@smestaens18j:/voletu/userlx/users/elaviole/Documents/L022-TP4_TP5/TD5$ splint prog3_corrected.c  
Splint 3.1.2 --- 04 Aug 2013  
Finished checking --- no warnings
```

L'analyse par l'outil Splint du programme corrigé ne présente plus de warning. Ainsi, nous pouvons voir que la correction proposée du programme a été efficace.

POINT DE VUE SUR CES TYPES D'OUTILS

Ces outils nous paraissent utiles afin de vérifier la validité d'un programme au niveau de la syntaxe. Cependant, nous avons noté que ces outils ne servent pas à vérifier le côté "logique" d'un programme. Ces outils ne permettent pas, par exemple dans le cas du programme 2, de nous informer que ce programme n'effectue rien de spécial, s'agissant de deux boucles "for" avec aucune instructions associées à celles-ci.