

TD1 :

Installation et configuration de vs sous vm.

Test du debugger

1) argc = 2 car on a 2 arguments : l'exécutable, et l'argument rentré (456).

argv[0] = / , car il correspond au 1er caractère du chemin de l'exécutable.

Donc le "résultat" est cohérent avec le contenu de launch.json.

47 est le code ascii du caractère /

2) prog = "../td01/debugSimple".

arg = 4 qui correspond au premier caractère de 456 (argument passé en paramètre).

3) &w : l'adresse du pointeur w.

*&w : la valeur de w, donc l'adresse de z (car *w = z).

**&w : la valeur de l'adresse pointée par *w, donc la valeur de z.

4) La sortie affiche le print (donc le "bonjour").

5) On s'arrête bien à la ligne 10.

6) argv[0] = "../debugSimple", chemin de debugSimple

argv[1] = 456, argument passé dans la ligne de commande

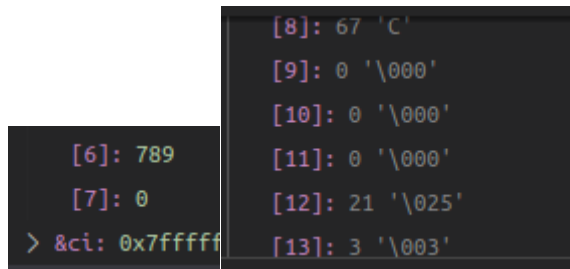
argv[2] = 0x0, pointe vers une zone mémoire vide

7) On obtient la liste des registres utilisés

```
-exec info registers
rax      0xfffffffffe175      140737488347509
rbx      0x55555555551e0      93824992236000
rcx      0x55555555551e0      93824992236000
rdx      0xffffffffd0de0      140737488346592
rsi      0xffffffffd0dc8      140737488346568
rdi      0x2                  2
```

8) 3. Un int est codé sur 8 octets, par rapport à un short qui n'est codé que sur 4 octets. Pour un short les 4 premiers sont utilisés, nous sommes ainsi en Little Endian.

4. Au début, la valeur et l'adresse sont les mêmes (pour 12,45 et 67). Pour la dernière valeur (789), trop longue pour un char, il y a une différence.



Index	Value	Hex
[8]	67	'C'
[9]	0	'\000'
[10]	0	'\000'
[11]	0	'\000'
[12]	21	'\025'
[13]	3	'\003'

5. ++ décale le pointeur en fonction de son type : pour un int ça sera de 4 cases, de 2 cases pour un short, et d'une case pour un char. Le pointeur ne sera donc pas décalé du bon nombre de cases si on n'a pas un int, ce qui explique les différences.

9) Le problème vient du fait que le programme veut faire $t[10] = 0$, alors que la dernière case est $t[9]$.

On remplace le \leq par un $<$.

10) L'algorithme de tri implémenté est shellsort.

11) Comme précédemment, on va un indice trop loin, ce qui cause le problème.

12) Lorsqu'on appelle mysort, il faut remplacer argc par $\text{argc}-1$ pour corriger le problème.

13) Le problème vient du fait qu'on veut faire un next sur quelque chose qui est null dans InsertATail.

Il faut simplement vérifier que le paramètre n'est pas nul avant de continuer.

14) La commande ltrace utilise la bibliothèque libc ainsi que les fonctions malloc, atoi, printf, putchar et free.

15) La bibliothèque se trouve dans `/lib/x86_64-linux-gnu/`

16)