

☛ Détecter les fuites mémoires en C avec Valgrind

Valgrind est un petit utilitaire qu'on peut lancer depuis un terminal pour analyser un exécutable C, notamment pour analyser quel usage de la mémoire fait le programme, et détecter d'éventuelles fuites mémoire. Après avoir compilé le programme avec gcc avec l'option -g, on peut lancer une exécution en demandant à Valgrind d'analyser ce qui se passe.

```
gcc -g mon_programme.c -o mon_programme
valgrind -v --leak-check=summary ./mon_programme
```

Cette opération produit beaucoup d'affichages^a, mais la partie HEAP SUMMARY est particulièrement intéressante puisqu'elle résume l'utilisation du tas. Ainsi, sur un programme sans fuite mémoire on obtient par exemple :

```
==8112== HEAP SUMMARY:
==8112==      in use at exit: 0 bytes in 0 blocks
==8112==    total heap usage: 11 allocs, 11 frees, 1,164 bytes allocated
==8112==
==8112== All heap blocks were freed -- no leaks are possible
```

tandis que sur un programme où des zones mémoires n'ont pas été libérées on obtient par exemple :

```
==8175== HEAP SUMMARY:
==8175==      in use at exit: 140 bytes in 10 blocks
==8175==    total heap usage: 11 allocs, 1 frees, 1,164 bytes allocated
==8175==
==8175== Searching for pointers to 10 not-freed blocks
==8175== Checked 74,744 bytes
==8175==
==8175== LEAK SUMMARY:
==8175==    definitely lost: 32 bytes in 2 blocks
==8175==    indirectly lost: 108 bytes in 8 blocks
==8175==    possibly lost: 0 bytes in 0 blocks
==8175==    still reachable: 0 bytes in 0 blocks
==8175==    suppressed: 0 bytes in 0 blocks
```

NB : Pour l'installation, vous pouvez vous référer à la page suivante.
<https://doc.ubuntu-fr.org/valgrind#installation>.

^a. il est sûrement possible de réduire ces affichages avec des options bien choisies