

LEPL1503/LSINC1503 - Cours 5

O. Bonaventure, B. Legat, L. Metongnon

Full Width Mode Present Mode

☰ **Table of Contents**

Rappels

- Lire [la partie multithread du syllabus](#)

Chacun doit contribuer sur le Git

- Pour valider votre contribution au projet, il est nécessaire que vous ayez des merge requests en votre nom qui sont mergées avec des commits également en votre nom
- Ces merge requests doivent avoir été finies (donc passer les tests) puis mergées
- Elles doivent apporter une contribution significatives au code, par exemple pas juste modifier le README ou reformatter
- L'excuse du peer coding (vous programmez à deux sur un ordinateur) n'est pas valide
- Ne pas savoir exécuter le code chez soi n'est pas une excuse. Git est facile à installer sous n'importe quelle plateforme et les tests s'exécute sur le runner GitLab.

Vérifiez qu'il n'y a pas de fuites de mémoires

Lors de la correction, nous vérifierons qu'il n'y a pas de fuites de mémoire avec [Valgrind](#)



```
#include <stdlib.h>

void f(void) {
    int* x = malloc(10 * sizeof(int));
    x[10] = 0; // problem 1: heap block overrun
} // problem 2: memory leak -- x not freed

int main(void) {
    f();
    return 0;
} // From https://valgrind.org/docs/manual/quick-start.html
```

① Compiling : `/home/runner/.julia/artifacts/e9aa081a3c28bdace800f92312f1212f6ec1921f/tools/clang /tmp/jl_B62PQF/main.c -o /tmp/jl_B62PQF/bin`

① Running : `valgrind /tmp/jl_B62PQF/bin`

```
==3524== Memcheck, a memory error detector
==3524== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==3524== Using Valgrind-3.22.0 and LibVEX; rerun with -h for copyright info
==3524== Command: /tmp/jl_B62PQF/bin
==3524==
==3524== Invalid write of size 4
==3524==   at 0x10915A: f (in /tmp/jl_B62PQF/bin)
==3524==   by 0x109183: main (in /tmp/jl_B62PQF/bin)
==3524==  Address 0x4a79068 is 0 bytes after a block of size 40 alloc'd
==3524==   at 0x4846828: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
==3524==   by 0x109151: f (in /tmp/jl_B62PQF/bin)
==3524==   by 0x109183: main (in /tmp/jl_B62PQF/bin)
==3524==
==3524==
==3524== HEAP SUMMARY:
==3524==   in use at exit: 40 bytes in 1 blocks
==3524== total heap usage: 1 allocs, 0 frees, 40 bytes allocated
==3524==
==3524== LEAK SUMMARY:
==3524==   definitely lost: 40 bytes in 1 blocks
==3524==   indirectly lost: 0 bytes in 0 blocks
==3524==   possibly lost: 0 bytes in 0 blocks
==3524==   still reachable: 0 bytes in 0 blocks
```

► Pourquoi valgrind ne donne-t-il pas le numéro de ligne ?

Interface par fichiers

On ne partage normalement pas de `.o` par `git` car ce n'est pas portable (e.g., ce `.o` ne marche pas sur Mac OS ni sur le Raspberry). Lorsque vous avez implémenté ces fonctions, supprimez ce fichier du `git` et ajoutez `objects/file.o` dans le `.gitignore`

Comme vous n'aurez pas vu les fichiers au moment de commencer le projet, nous vous fournissons un fichier objet `file.o` dans le dossier `objects`. ... Cependant, vous devrez coder ces fonctions plus tard et ceci sera vérifié !

C'est une bonne chose d'écrire des unit tests mais comme les signatures des fonctions sont libres, nous utiliseront l'interface uniformisée par fichier pour tester votre code.

Le Makefile génère un exécutable `main`. Vous pouvez l'utiliser de la façon suivante :

```
./main name_op input_file_A [-v] [-n n_threads] [-f output_stream] [-d degree] [input_file_B]
```



```
Activating project at '~/work/LEPL1503/LEPL1503/Lectures'
```

