

Enseignant(s)

REY Anthony

Email(s)

me@anghr.com**Convertisseur d'images multi-threadé****1** Matières, formations et groupes

Matière liée au projet :

Formations : **5ESGI - Groupe AL2 ||**Nombre d'étudiant
par groupe :**2**Règles de constitution des groupes: **Libre**Charge de travail
estimée par étudiant : **10,00 h****2** Sujet(s) du projetType de sujet : **Imposé****Convertisseur d'images multi-threadé**

Voir descriptif détaillé du projet

3 Détails du projet**Objectif du projet (à la fin du projet les étudiants sauront réaliser un...)**

Adapter un programme en version multithread

Descriptif détaillé

Transformer edge-detect.c en un programme multi-threadé respectant le cahier des charges suivant :

- Architecture producteur-consommateur
 - 1 consommateur (se sert dans la pile partagée et écrit les fichiers sur le disque)
 - N producteurs (convertissent les fichiers puis les mettent dans la pile partagée)
- Le code doit être vectorisé au possible (la partie conversion d'images)
 - Utiliser l'autovectorisation de gcc
- Le code doit fonctionner avec des bmp de plus de 100 mo (j'en ai dans mon jeu d'essai)
- Passer en parametre le dossier d'origine :
 - Non vide, sinon le programme se termine
 - On traite uniquement les fichiers .bmp
- Passer en parametre le dossier de destination :
 - Il est vidé quand le programme démarre
- Passer en parametre le nombre de threads producteurs voulus (< nb de fichiers dans le dossier d'origine && > 0) :
 - A l'initialisation des threads producteurs, pour faire simple, on leur passera la liste des fichiers a traiter
- Passer en parametre l'algorithme à appliquer sur chaque image :
 - On veut les suivants :
 - boxblur
 - edgedetect
 - sharpen

Si il manque un paramètre on affiche l'exemple suivant :
./apply-effect "./in/" "./out/" 3 boxblur

NB :

https://en.wikibooks.org/wiki/C_Programming/POSIX_Reference/dirent.h

Au début, essayez de faire le programme en utilisant des #define plutôt que la command-line.
Résolvez les problèmes spécifiques plus tard dans la conception (On traite uniquement les fichiers .bmp, non vide etc...)
Pour l'utilisation des différents algorithmes sur les images, il suffit juste de changer les valeurs de la matrice de convolution.

Une fois que vous avez un programme qui marche, attaquez vous aux problématiques plus spécifiques.
C'est pas une épreuve de programmation en C, mais je demande un minimum de recherche personnelle, auquel cas je serais indulgent si il manque des trucs.

Ouvrages de référence (livres, articles, revues, sites web...)

[https://fr.wikipedia.org/wiki/Noyau_\(traitement_d'image\)](https://fr.wikipedia.org/wiki/Noyau_(traitement_d'image))

Outils informatiques à installer

GCC 9

4 Livrables et étapes de suivi

1	Rendu final	Rendu final	dimanche 27/01/2019 23h00
---	-------------	-------------	---------------------------------

Durée de présentation
par groupe :

15 min

Audience : **A huis clos**

Type de présentation :

Présentation / PowerPoint - Démonstration

Précisions :

Présentation, decorticage, interrogatoire