

INF3135

Construction et maintenance de logiciels

Cours 7: Révision

Alexandre Blondin Massé

Université du Québec à Montréal
Département d'informatique

Été 2020

Table des matières

- 1 Travail pratique 1
- 2 Résumé du chapitre 4
- 3 Quiz 2
- 4 Exercices

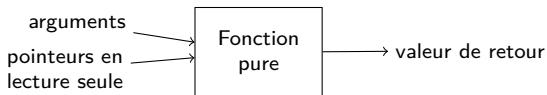
Travail pratique 1

Travail pratique 1

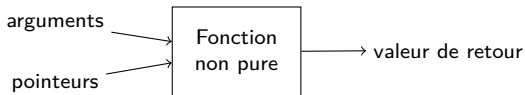
- Date de remise: **19 juin**, à **23h59**
- **20%** de la note totale
- Doit être fait **seul**
- **Dépôt GitLab**: doit être *forké*, sa visibilité mise à *privé*, puis l'accès en mode *Developer* doit être donné à blondin_al
- **Description du travail**: dans le fichier `sujet.md`
- À **compléter**: `canvascii.c`, `Makefile`, `.gitignore`, `README.md`
- À **modifier**: dans `check.bats`, supprimer `skip` pour activer le test

Résumé du chapitre 4

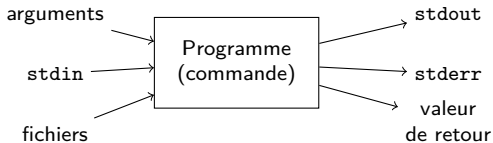
Entrées et sorties



Entrées



Sorties



La bibliothèque `stdio.h`

Macros:

- EOF: caractère de fin de fichier
- `stdin`, `stdout`, `stderr`: canaux standards
- NULL: pointeur nul, ...

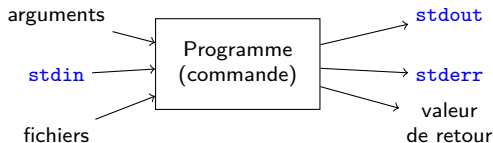
Types:

- FILE: flux (*stream*)
- `size_t`: taille en octets
- `fpos_t`: position dans un flux, ...

Variables externes: `optarg`, `opterr`, `optind`, `optopt` (gestion des arguments)

Plusieurs dizaines de **fonctions** (`man stdio`)

Canaux



3 canaux standards

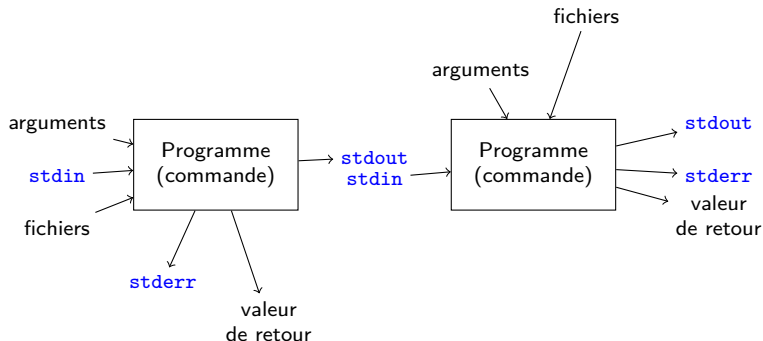
- `stdin`: entrée standard (canal 0)
- `stdout`: sortie standard (canal 1)
- `stderr`: sortie d'erreur standard (canal 2)

Redirections

- `commande < fichier`: redirige fichier sur `stdin`
- `commande > fichier`: redirige `stdout` dans fichier
- `commande 2> fichier`: redirige `stderr` dans fichier

Tubes

- Permet d'**enchaîner** des programmes
- Le contenu écrit sur stdout par la première commande
- Est lu sur stdin par la deuxième commande
- **Syntaxe:** commande1 | commande2 | ... | commandeN



Valeur de retour

Généralités

- 0: le programme s'est terminé « **normalement** »
- $\neq 0$: le programme s'est terminé « **anormalement** »
- Valeur de retour de la **dernière commande**
- Contenue dans la **variable spéciale** \$?

En C

- **Valeur** retournée par la fonction `main`
- Ou **argument** passé à la fonction `exit`

Combinaisons de commandes

- Avec `;` `&&` et `||`
- Valeur de retour **influence** `&&` et `||`

Commandes utiles

- **Graphviz**: <https://graphviz.org/>, pour générer des graphes, format DOT, en ligne de commande
- **QPDF**: <http://qpdf.sourceforge.net/>, pour manipuler des documents PDF
- **ImageMagick**: <https://imagemagick.org/index.php>, pour manipuler des images
- **FFMPEG**: <https://ffmpeg.org/>, pour manipuler des fichiers audio et vidéo
- **Gnuplot**: <http://www.gnuplot.info/>, pour générer des graphiques, des courbes, des surfaces, etc.

Quiz 2

Quiz 2

Contenu évalué

- **Chapitre 2:** Outils de développement logiciels
- **Chapitre 3:** Pointeurs
- **Chapitre 4:** Entrées et sorties
- Contenu des laboratoires

Forme: 1 heure 10, plage de 24 heures

- **3** questions courtes ou à choix multiples
- **1** question longue

Avant le quiz

Préparation

- Écouter/réécouter les **capsules**
- Réviser les **cours**
- Faire les **labos**
- Regarder les **solutions** des labos
- Attention à la **qualité**: généralité, efficacité, simplicité, lisibilité

Environnement

- Bonne **connexion** réseau
- **Environnement** de développement
- Terminal
- Débogueur si nécessaire
- Bien vérifier que vos réponses sont enregistrées
- Bien gérer votre temps

Exercices

Exercices

1. Écrivez un programme nommé `distr` qui lit sur l'entrée standard une **liste de mots** et qui écrit sur la sortie standard la **distribution de la longueur** des mots
2. Utilisez **Gnuplot** pour visualiser le résultat
3. Modifiez le programme précédent pour qu'il accepte **1 argument** correspondant à un nom du fichier à lire en entrée plutôt que sur `stdin`
4. Modifiez le programme précédent pour qu'il accepte **1 ou 2 arguments** correspondant à des noms de fichiers à lire en entrée et en sortie plutôt que sur `stdin` et `stdout`