

INF3135

Construction et maintenance de logiciels

Cours 4: Outils de développement logiciel

Alexandre Blondin Massé

Université du Québec à Montréal
Département d'informatique

Été 2020

Table des matières

1 Exercices

2 Travail pratique 1

3 Résumé des capsules

4 Quiz 1

Exercices

Exercices

1. Écrire un programme qui
 - définit un **type** `struct square_matrix` permettant de représenter une matrice carrée de doubles (vous pouvez supposer que l'ordre de la matrice est limité à 10);
 - définit une **fonction** `initialize_matrix(m,n,v)` qui crée une matrice `m` de dimensions `n x n` et qui initialise chacune des entrées de la matrice à `v`;
 - définit une **fonction** `print_matrix(m)` permettant d'afficher le contenu de la matrice sur `stdout`.
2. Écrire un programme nommé `somme.c` qui prend exactement un argument de la forme `a,b,c`, où `a`, `b` et `c` sont des entiers, et qui affiche la somme des trois nombres sur `stdout`:

```
$ gcc -o somme somme.c
```

```
$ ./somme 1,2,3
```

```
6
```

Travail pratique 1

Résumé des capsules

Style de programmation

- C a été **standardisé** dans les années 80 (ANSI C89/C90)
- Mais aucun standard de **programmation** proposé

Quelques exemples

- Indian Hill
- NASA
- Noyau Linux (Linus Torvalds)
- GNU
- GNOME

« The Single Most Important Rule »

« Check the surrounding code and try to imitate it. »

— Extrait du site de GNOME

Documentation

Code source (*docstrings*)

- **Modules:** en-tête de fichier
- **Types:** sémantique des types
- **Fonctions:** description, paramètres et valeur de retour

Utilisateur

- **Manuel d'utilisation:** fichier README
- Bien exploiter le format Markdown

Bats

```
@test "Addition" {  
    resultat="$(echo $((1 + 2)))" # Pas obligé d'utiliser run  
    [ "$resultat" -eq 3 ]         # Teste si sortie de echo est 3  
}  
  
@test "Avec run" {  
    run echo $((1 + 2))           # Avec run  
    [ "$status" -eq 0 ]          # Teste code de retour de echo  
    [ "$output" == "3" ]         # Teste si sortie de echo est 3  
}  
  
@test "Plusieurs lignes" {  
    run echo -e "ligne 1\nligne 2" # Avec run  
    [ "${lines[0]}" == "ligne 1" ] # Teste contenu de la 1re ligne  
    [ "${lines[1]}" == "ligne2" ]  # Teste contenu de la 2e ligne  
}  
  
@test "Un test désactivé" {  
    skip                          # On désactive le test  
    run echo "un autre test"  
    [ "$status" -eq 1 ]          # Teste si echo échoue  
}
```

Git

- `git log`: voir l'historique
- `git gr`: voir le graphe de l'historique
- `git init`: initialiser un dépôt
- `git clone`: copier un dépôt existant
- `git st`: voir l'état du projet
- `git checkout`: changer l'état du projet
- `git diff`: voir les différences
- `git show`: montrer un *commit*
- `git add`: indexer des modifications
- `git commit`: valider des modifications
- `git reset`: réinitialiser l'état d'un projet
- `git remote`: gérer des dépôts distants
- `git fetch`: télécharger des références
- `git pull`: récupérer des modifications
- `git push`: partager des modifications

GitLab-CI

```
# Mise à jour de apt et installation de Bats
before_script:
  - apt-get update -qq
  - git clone "https://github.com/bats-core/bats-core.git" /tmp/bats
  - mkdir -p /tmp/local
  - bash /tmp/bats/install.sh /tmp/local
  - export PATH="$PATH:/tmp/local/bin"

# Pour vérifier la compilation
build:
  stage: build
  script:
    - make

# Tests unitaires
test:
  stage: test
  script:
    - make test
```

- Possible de spécifier l'**image** Docker
- Structuration des *pipelines*
- Peut conserver les **résultats** (artéfacts)

Quiz 1

Quiz 1

Contenu

- **3** questions courtes ou à choix multiples
- **1** question à développement (compléter un programme)
- Entre 26 mai **20h00** et 27 mai **23h00**
- Durée: **1 heure**
- **Lien Moodle**

« Règles »

- Droit à documentation **écrite**
- Droit de lire des trucs sur le *web*
- Droit de tester des programmes sur ordi
- Interdiction de **discuter** avec autres étudiants
- Interdiction de **demande de l'aide** en ligne ou en personne
- Assurez-vous d'avoir une **bonne connexion**
- Assurez-vous d'avoir un **environnement de développement** en C