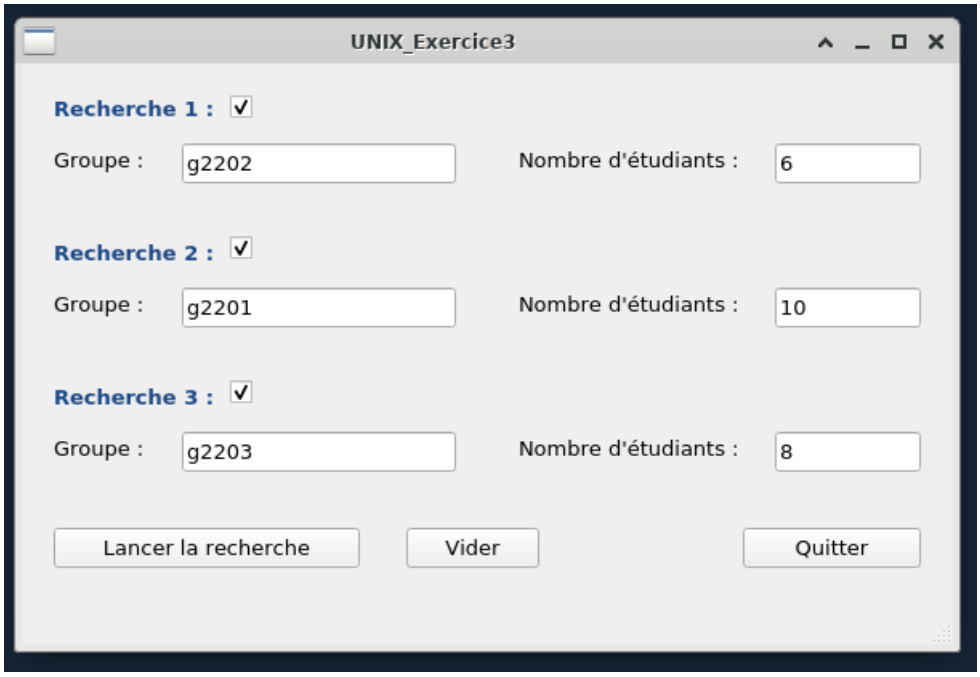
Exercice n°3 : la création des processus

Objectifs :

- Se familiariser avec la création de processus (**fork**, **exec**, **exit** et **wait**).
- Se familiariser avec l'utilisation de la librairie C d'accès à une base de données MySql.
- Création d'un fichier de traces.
- Et toujours : création d'un makefile

Description générale :

L'application ressemble visuellement à



Elle permet de faire une ou plusieurs recherches en base de données et d'y récupérer le nombre d'étudiants du ou des groupes dont on précise le nom dans la partie de gauche de la fenêtre. Les résultats de la recherche s'affichent dans la partie droite de la fenêtre.

La fenêtre graphique a été créée pour vous et vous est fournie sous la forme d'un ensemble de fichiers sources. De même, un programme créant la base de données et un autre réalisant la recherche dans la base de données vous sont fournis.

Etape 1 : Création d'un fichier makefile

On vous fournit une archive **UNIX_Exercice3.tar** qui contient les fichiers suivants :

- **main.cpp** : le main de votre application
- **mainwindowex3.cpp** : code source de la fenêtre de l'application, seul fichier que vous devez modifier.
- **Mainwindowsex3.h** : fichier header correspondant au fichier mainwindow.cpp.

- **moc_mainwindowex3.cpp** et **ui_mainwindowex3.h** : fichiers sources propres à Qt nécessaires à la fenêtre graphique.
- **CreationBD.cpp** : programme qui crée la table UNIX_EX3 dans la base de données et qui y ajoute un ensemble d'étudiants de différents groupes.
- **Lecture.cpp** : programme qui permet de faire une recherche dans la table UNIX_EX3 de la base de données.
- **Compile.sh** : fichier texte (on appelle ça un script) contenant les lignes de compilation nécessaires pour la création des fichiers exécutables « UNIX_Exercice3 », « CreationBD » et « Lecture ».

Il est déjà possible de compiler l'application en utilisant le fichier **Compile.sh** :

```
# sh Compile.sh
# UNIX_Exercice3
```

Le but est donc de créer un fichier **makefile** permettant de réaliser la compilation de tous les fichiers nécessaires à la création de l'exécutable **UNIX_Exercice3** mais également des deux exécutables **CreationBD** et **Lecture**.

Etape 2 : Creation de la table UNIX_EX3 et accès à MySql

Une base de données a déjà été créée pour vous sur

- La machine virtuelle moon (BD = **PourStudent** ; login = **Student** ; password = **PassStudent1_**)
- La machine de l'école jupiter (BD = **PourXXX** ; login = **XXX** ; password = **PassXXX1_**) où **XXX** est votre login (tout en minuscule).

La création de la table UNIX_EX3 se fait à l'aide de l'exécutable **CreationBD** que

1. **vous ne devez pas modifier**,
2. vous devez comprendre sur le principe (connexion à la BD, exécution d'une requête).

Pour cela, il suffit de taper la commande

```
# CreationBD
```

Pour vérifier que la table a été créée correctement, voici un exemple de manipulation de MySql :

```
# mysql -u Student -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 48
Server version: 8.0.21 Source distribution

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| PourStudent        |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.01 sec)

mysql> use PourStudent;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----------------------+
| Tables_in_PourStudent |
+-----------------------+
| UNIX_EX3              |
+-----------------------+
1 row in set (0.00 sec)

mysql> select * from UNIX_EX3;
+----+--------+--------+
| id | nom    | groupe |
+----+--------+--------+
|  1 | aaa010 | g2201  |
|  2 | ccc002 | g2203  |
|  3 | bbb001 | g2202  |
|  4 | ddd005 | g2204  |
|  5 | aaa006 | g2201  |
...
| 30 | ccc006 | g2203  |
+----+--------+--------+
30 rows in set (0.01 sec)

mysql> select count(*) from UNIX_EX3 where groupe = 'g2202';
+----------+
| count(*) |
+----------+
|        6 |
+----------+
1 row in set (0.00 sec)

mysql> exit
Bye
#
```

L'exécutable **Lecture** (**dont vous ne devez pas modifier le code mais le comprendre dans les grandes lignes**) reçoit en argument le nom du groupe, fait la recherche dans la base de données et retourne (via un appel de exit) le nombre d'étudiants de ce groupe :

```
# Lecture
Erreur: Trop ou trop peu d'argument(s)...
```

```
# Lecture g2202
Lecture pour le groupe g2202
(7738) g2202  (attend 6 secondes)
# echo $?
6
#
```

L'exécution de la commande « **echo $?** » permet d'afficher la valeur de l'**exit** du dernier processus qui a été lancé en ligne de commande.

Etape 3 : Création d'un processus fils

Dans un premier temps, une seule recherche (pour un seul groupe) sera réalisée. L'appui sur le bouton « Lancer la recherche »

1. créera un processus fils (utilisation de **fork**) ;
2. le fils exécutera **Lecture** (utilisation de **exec**) en recevant en argument le nom du groupe correspondant à la recherche 1 de la fenêtre ;
3. le processus père attendra la fin de son fils (utilisation de **wait**) et affichera la valeur de l'exit de son fils dans la partie droite de la fenêtre.

Dans cette partie, vous ne devez modifier que la fonction « **on_pushButtonLancerRecherche_clicked** » de la classe MainWindowEx3.

Etape 4 : Création de plusieurs processus fils

Le principe reste le même que dans l'étape 3. On lance 0, 1, 2 ou 3 processus en fonction que les checkbox sont cochés ou non. Pour savoir si un checkbox a été coché par l'utilisateur, vous pouvez utiliser les méthodes

- **bool MainWindowEx3::recherche1Selectionnee()**
- **bool MainWindowEx3::recherche2Selectionnee()**
- **bool MainWindowEx3::recherche3Selectionnee()**

qui retournent true si le checkbox correspondant est sélectionné et false sinon.

Les 1, 2 ou 3 processus fils doivent être créés (**fork** + **exec**) et **tourner en parallèle**. Le processus père doit attendre la fin de ses 1, 2 ou 3 fils (**wait**) et afficher leur valeur de retour dans la case correspondante de la partie droite de la fenêtre. A nouveau ici, tout se passe dans la fonction « **on_pushButtonLancerRecherche_clicked** » de la classe MainWindowEx3.

Vous à présent implémenter les boutons

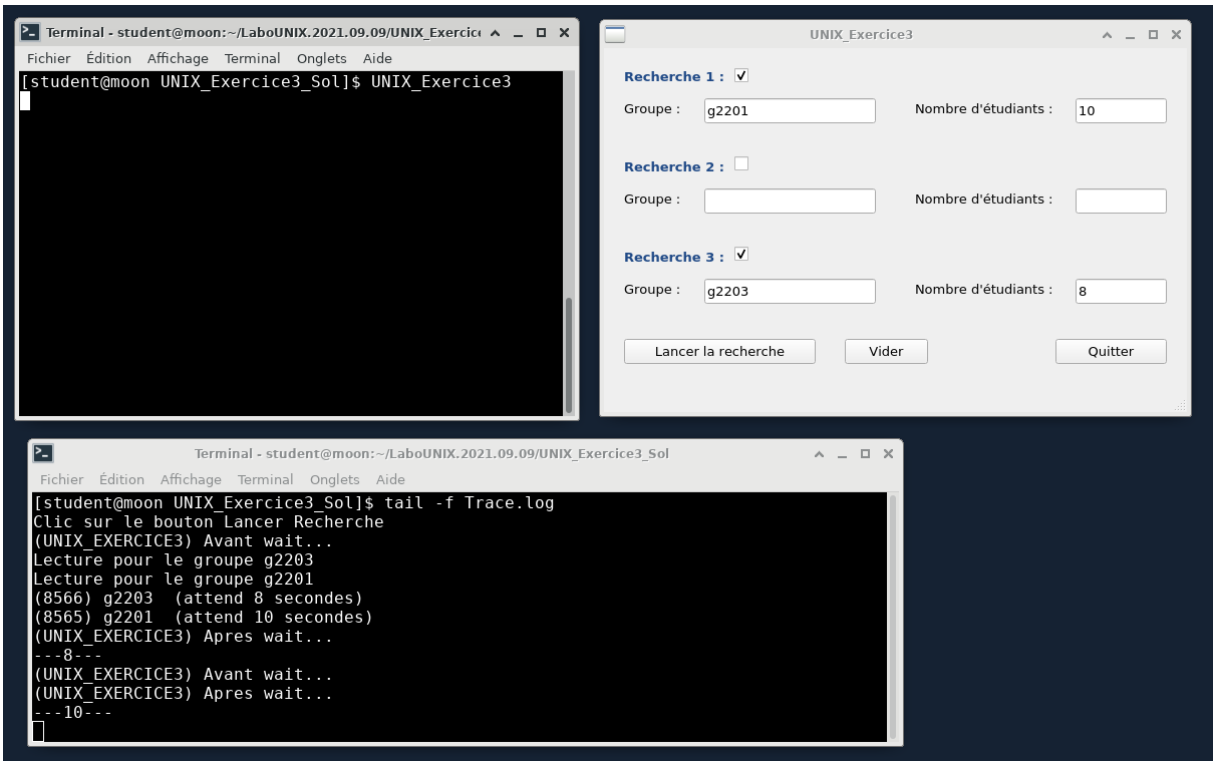
- « Vider » dont l'action est de vider les 6 champs de texte de la fenêtre.
- « Quitter » dont l'action est de quitter proprement l'application.

Etape 5 : Création d'un fichier de traces

On vous demande de rediriger la sortie standard d'erreur (stderr) vers un fichier texte appelé « **Trace.log** » (utilisation de **open** et **dup/dup2**).

Processus père et fils écriront tous dans le même fichier.

Il est possible de suivre en direct l'évolution de l'écriture dans le fichier de traces en utilisant la commande **tail** :



Bon travail !

Remarque

Dans l'état actuel des choses, les programmes CreationBD et Lecture sont configurés pour fonctionner sur la machine virtuelle moon. Lorsque vous ferez tourner l'application sur la machine jupiter de l'école, n'oubliez pas de modifier le nom de la base de donnée, le user et le password dans CreationBD.cpp et Lecture.cpp.