

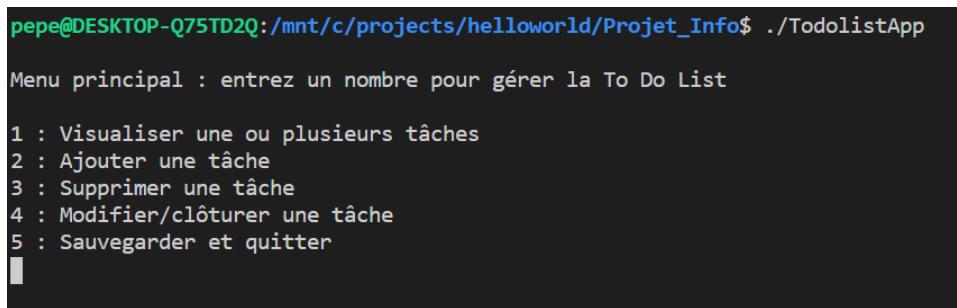
Doc programme Todolist

Louis Blazejczak

31 mars 2020

1 Fonctionnalités

J'ai codé ce programme dans l'objectif d'avoir quelque chose un minimum "interactif", donc il suffit de lancer le logiciel et celui-ci gère tous les appels de fonctions. L'utilisateur a juste à input des données via `std::cin` et `getline()`. Le programme boucle tant que l'utilisateur n'en demande pas la sortie (cf. capture d'écran du menu principal ci-dessous : il faut simplement entrer '5' dans l'invite de commande).



```
pepe@DESKTOP-Q75TD2Q:/mnt/c/projects/helloworld/Projet_Info$ ./TodolistApp
Menu principal : entrez un nombre pour gérer la To Do List
1 : Visualiser une ou plusieurs tâches
2 : Ajouter une tâche
3 : Supprimer une tâche
4 : Modifier/clôturer une tâche
5 : Sauvegarder et quitter
```

L'utilisation est donc assez intuitive. Le programme remplit EN THÉORIE toutes les fonctionnalités décrites par le sujet, mais il y a un mais (voir suite).

Pour ce qui est du code, j'ai essayé de commenter chaque fonction avec une petite spécification, et parfois le pourquoi du comment de la fonction. J'espère que vous trouverez ce code suffisamment lisible.

2 Problèmes & bugs

2.1 `std::basic_string::_M_construct null not valid`

Voici l'erreur la plus coriace que j'ai rencontrée, et mon interprétation du problème.

Le plus gros problème de cette to do list est qu'elle ne fonctionne que partiellement. En effet, la première erreur que l'utilisateur constatera lors de l'utilisation est que lorsqu'il ajoute une sous-tâche et quitte le programme, une exception

`std::logic_error : basic_string::_M_construct null not valid` est `throw`. J'ai essayé à maintes reprises de corriger ce bug mais je n'ai pas réussi à faire fonctionner le debugger sous VSCode (c'est le debugger Python qui se lance), donc j'ai opté pour une méthode plus barbare. J'ai essayé d'identifier où avait lieu l'erreur en insérant des `std::cout <<` à quasiment toutes les lignes du corps de la fonction `TaskManager::save()` pour voir où était l'erreur.

```

int id = t->id;
// std::cout << "id check" << id << '\n';
std::string title = t->title;
// std::cout << "title check" << title << '\n';
std::string description = t->description;
// std::cout << "desc check" << description << '\n';
std::string d_creation = t->d_creation;
// std::cout << "d_crea check" << d_creation << '\n';
std::string d cloture = t->d cloture;
// std::cout << "d_clot check" << d cloture << '\n';
std::string status = t->status;
// std::cout << "status check" << status << '\n';
int progress = t->progress;
// std::cout << "progress check" << progress << '\n';
int priority = t->priority;
// std::cout << "prio check" << priority << '\n';
std::string comms = t->comms;
// std::cout << "comm check" << comms << '\n';

```

Vu la nature de l'exception, je m'attendais à avoir maladroitement déclaré une variable de type `std::string` avec une `std::string` qui avait déjà été `delete` ou qui était `out of scope`. Par ce procédé, j'ai plus ou moins identifié d'où venait cette erreur : lorsque j'enregistre une `subtask`, elle est enregistrée deux fois : en tant que `subtask`, et en tant que `task` à part entière (ce qui est voulu!). Mais je pense que comme je fais appel deux fois aux attributs d'une `task` (voir screen ci-dessous : il faudrait que j'appelle en plus `std::vector::erase()`...) lors de l'appel de `TaskManager::save()`, les attributs de type `std::string` sont détruits.

5 appel de `TaskManager::save()`

```

id check 0
title checkTITRE
desc checkDESCRIPTION TACHE 1
d_crea checkTue Mar 31 13:23:07 2020
d_clot checkTâche non terminée
status checkEn cours
progress check50
prio check5
comm checkCOMMENTAIRES ICI
id check 1 accès correct aux attributs de la tâche 1 (en tant que tâche)
title checkSOUS TACHE
desc checkSOUS TACHE DESCRIPTION
d_crea checkTue Mar 31 13:23:06 2020
d_clot checkTâche non terminée
status checkEn cours
progress check60
prio check3
comm checkSOUS TACHE COMMENTAIRES à ce stade la task 0 a été delete !
id check 0 spaghetti code : il reste la tâche 0 dans le vecteur des tâches. Mais comment la supprimer ??? Lorsque
title check j'appelle std::vector::erase() après TaskManager::write_task(), j'ai une seg fault ...
terminate called after throwing an instance of 'std::logic_error'
  what(): basic_string::_M_construct null not valid
Aborted (core dumped)

```

Tous les attributs ont été correctement accédés pour la tâche dont l'id est 0 et qui a 1 comme subtask

à ce stade la task 0 a été delete !

spaghetti code : il reste la tâche 0 dans le vecteur des tâches. Mais comment la supprimer ??? Lorsque j'appelle std::vector::erase() après TaskManager::write_task(), j'ai une seg fault ...

basic_string::_M_construct null not valid

Je n'ai pas réussi à trouver de solution...

Donc, tant qu'il n'y a pas de sous-tâches dans la todolist, le programme fonctionne. Dès qu'il y en a une, il plante (en fait ce n'est pas systématique, et c'est ça que je ne comprends pas)... Encore plus drôle (si on veut), comme dans l'exemple précédent le programme a eu le temps d'écrire dans le fichier de sauvegarde, le programme se lance correctement en chargeant les données du fichier, puis se sauvegarde correctement, sans erreur, alors que la structure tâche/sous-tâche est conservée ! Donc, je ne suis pas sûr que mon interprétation soit exacte, mais elle m'a semblé plausible.

2.2 Fonctionnalités manquantes

Certaines fonctionnalités sont manquantes pour ce programme : les sous-tâches ne peuvent pas avoir de sous-tâches, une option pour éviter de ré-afficher les sous-tâches clôturées, mais non

supprimées (donc celles qui ont une date de clôture différente de celle par défaut), la possibilité de modifier en place les attributs (notamment le commentaire d'une tâche) entre autres.

Et SURTOUT : le manque de sécurité vis-à-vis d'inputs utilisateur douteux. J'ai ajouté ici et là quelques sécurités, du type qui refusent un input non entier, mais ce n'est pas généralisé à tout le code. Le programme ne fonctionne donc QUE en utilisation "normale" (et sans sous-tâches). Idem pour les caractères "délimiteurs" (cf le fichier README) qui ne sont pas refusés à l'input et qui font faire n'importe quoi au programme.

Enfin, la fonctionnalité de suppression de tâche est buggée lorsqu'on supprime une tâche qui est sous-tâche d'une autre (pourtant, j'ai essayé de gérer cet aspect dans le code...)

2.3 using namespace std;

Cette ligne de code m'a posé beaucoup de problèmes sans que je le sache, du fait de l'utilisation de `fstream`, `string`, `cin`, `getline()`, etc. Je rencontrais très souvent l'erreur `X is ambiguous` avec `X` une variable, une fonction, un type...

Je me suis donc débarrassé de cette ligne et mis des `std::` partout, ce qui rend le code moins lisible je trouve. J'espère que vous vous y retrouverez.

3 Conclusion

C'est déjà la fin de cette "doc" (qui finalement est un répertoire de ce qui ne va pas dans ce programme), puisque le fonctionnement est assez intuitif. Je mets tout de même en fin de document des exemples d'utilisation.

Je pense que la plus grande difficulté que j'ai rencontrée est d'avoir été occupé pendant le stage ouvrier, j'ai donc repris mon code que j'avais écrit avant le stage, dont j'avais quasiment tout oublié. C'est pourquoi il y a des fonctions dont je ne me sers quasiment pas (la plupart des fonctions membres de la classe `task`, en fait) mais j'ai peur de les supprimer, parce qu'elles servent peut-être quelque part, et donc je devrais les remplacer. Désolé pour ces lacunes, et bien sûr désolé puisque le programme ne marche pas, ce qui est plutôt gênant.

4 Illustrations

```
Menu principal : entrez un nombre pour gérer la To Do List
1 : Visualiser une ou plusieurs tâches
2 : Ajouter une tâche
3 : Supprimer une tâche
4 : Modifier/clôturer une tâche
5 : Sauvegarder et quitter
2
Saisissez l'intitulé de la tâche : Chercher courses mamie
Description de la tâche : Au drive
Avancement actuel (pourcentage de 0 à 100, sans le caractère %) : 40
Priorité (int) : 7
Commentaires : Prendre masque et gants !
Ajouter des sous-tâches ? [Y/N] y
[C]réer une nouvelle tâche ou [A]jouter une tâche existante comme sous-tâche ? [C/A]
c
```

```

[C]réer une nouvelle tâche ou [A]jouter une tâche existante comme sous-tâche ? [C/A]
c
Saisissez l'intitulé de la tâche : Protection corona
Description de la tâche : Acheter masque et gants
Avancement actuel (pourcentage de 0 à 100, sans le caractère %) : 0
Priorité (entier) : 10
Commentaires : Si possible un stock!
Ajouter une autre tâche ? [Y/N] n
Sous-tâches enregistrées

Menu principal : entrez un nombre pour gérer la To Do List

1 : Visualiser une ou plusieurs tâches
2 : Ajouter une tâche
3 : Supprimer une tâche
4 : Modifier/clôturer une tâche
5 : Sauvegarder et quitter

1
Veuillez entrer les identifiants des tâches que vous souhaitez visualiser, séparés par un espace. Pour tout visualiser, tapez ALL
ALL
1 : Protection corona
  Description :
    Acheter masque et gants

  Commentaires :
    Si possible un stock!

  Créeée le : Tue Mar 31 15:52:01 2020
  Statut : En cours, avancement : 0, priorité : 10
  Tâches préalables nécessaires : Aucune
  Tâche non terminée

0 : Chercher courses mamie
  Description :
    Au drive

  Commentaires :
    Prendre masque et gants !

  Créeée le : Tue Mar 31 15:52:03 2020
  Statut : En cours, avancement : 40, priorité : 7

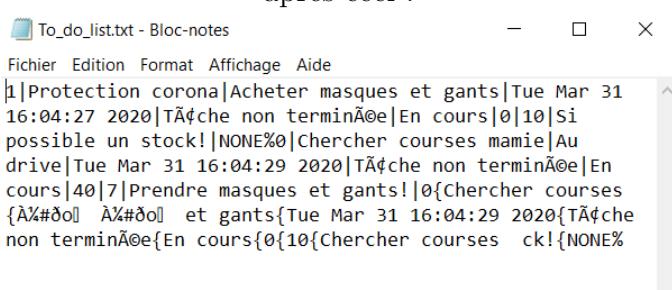
  Tâches préalables nécessaires : 1
  Tâche non terminée

  Menu principal : entrez un nombre pour gérer la To Do List

  1 : Visualiser une ou plusieurs tâches
  2 : Ajouter une tâche
  3 : Supprimer une tâche
  4 : Modifier/clôturer une tâche
  5 : Sauvegarder et quitter
  5
  Tâches enregistrées avec succès

```

D'ailleurs dans cet exemple, l'enregistrement fonctionne. Voici ce qu'il y a dans le fichier après ceci :



Vu la tête des derniers caractères, on comprend mieux que le programme bug. Mais d'où viennent ces caractères : aucune idée.

Le programme recharge bien les données par la suite :

```
pepe@DESKTOP-Q75TD2Q:/mnt/c/projects/helloworld/Projet_Info$ ./TodolistApp
```

```
Menu principal : entrez un nombre pour gérer la To Do List
```

```
1 : Visualiser une ou plusieurs tâches  
2 : Ajouter une tâche  
3 : Supprimer une tâche  
4 : Modifier/clôturer une tâche  
5 : Sauvegarder et quitter
```

```
1
```

```
Veuillez entrer les identifiants des tâches que vous souhaitez visualiser, séparés par un espace. Pour tout visualiser, tapez ALL
```

```
ALL
```

```
1 : Protection corona
```

```
Description :  
    Acheter masques et gants
```

```
Commentaires :
```

```
    Si possible un stock!
```

```
Créée le : Tue Mar 31 16:04:27 2020
```

```
Statut : En cours, avancement : 0, priorité : 10
```

```
0 : Chercher courses mamie  
Description :  
    Au drive
```

```
Commentaires :  
    Prendre masques et gants!
```

```
Créée le : Tue Mar 31 16:04:29 2020
```

```
Statut : En cours, avancement : 40, priorité : 7
```

```
Tâches préalables nécessaires : 0
```

```
Tâche non terminée
```

```
Menu principal : entrez un nombre pour gérer la To Do List
```