



Outils de développement logiciel

TP n° 8 : tests unitaires et compilation multifichiers

Alain Lebreton

2023-2024

Objectif

Tester ses fonctions.

Pré-requis : chapitres 9 et 11

Durée estimée : 1 séance

1 Introduction

La ration calorique exprime le nombre de calories nécessaires à un individu pendant 24 heures. Elle correspond au produit du métabolisme de base par un facteur d'activité. Les lignes qui suivent présentent une façon simplifiée de la calculer.

Métabolisme de base

Le métabolisme de base est le nombre de calories nécessaires à l'organisme pendant 24 heures (si le sujet restait couché).

La formule de calcul du métabolisme de base (MB) donnée dans [1] a été simplifiée et se présente à présent sous la forme :

$$MB = 24 \times \alpha \times SC$$

où SC est la surface corporelle telle que :

$$SC = 0.2 \times T^{0.7} \times P^{0.4} \text{ (voir [2])}$$

avec P le poids (kg), T la taille (m) et α un coefficient qui dépend de l'âge (les enfants de moins de 10 ans ne seront pas considérés) :

- $\alpha = 45$ de 10 à 14 ans ;
- $\alpha = 40$ de 15 à 19 ans ;
- $\alpha = 35$ de 20 à 60 ans ;
- $\alpha = 40$ au-delà de 60 ans.

De plus, une femme, à taille, poids et âge égaux, a un métabolisme de base inférieur de 10 % à celui d'un homme. Par contre, une femme enceinte ou qui allaite doit absorber un supplément d'environ 600 calories.

Facteur d'activité

Le facteur d'activité (FA) dépend de l'activité physique du sujet pendant la semaine. Cinq types d'activités hebdomadaires sont décrits dans [3] :

1. Aucune à très petite (sédentaire) : $FA = 1,2$.
2. Légère (1 à 3 jours d'exercices) : $FA = 1,375$.
3. Modérée (3 à 5 jours d'exercices) : $FA = 1,55$.
4. Importante (6 à 7 jours d'exercices) : $FA = 1,725$.
5. Très importante (exercices intensifs deux fois par jour) : $FA = 1,9$.

2 Organisation du travail

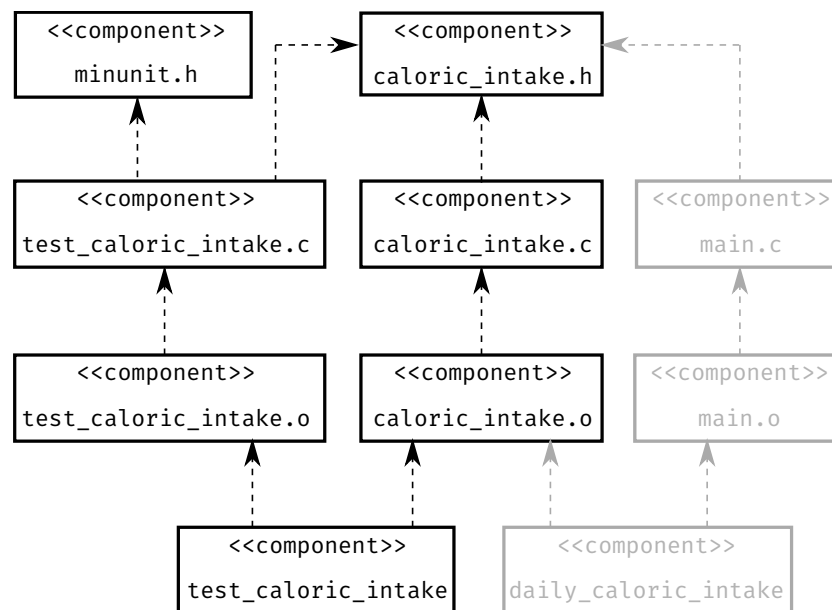
La réalisation de l'application permettant de calculer les rations caloriques journalières doit se faire en utilisant un développement **itératif** et **incrémental** avec **validation**. C'est-à-dire que les fonctions doivent en principe être testées dès leur écriture et séparément du programme global. Ici, nous trichons un peu en vous fournissant déjà les fonctions de calcul qui malgré tout peuvent encore comporter quelques défauts.

La validation sera réalisée en mettant en oeuvre la bibliothèque **MinUnit** contenue dans l'unique fichier d'entête "minunit.h" présent dans le dossier "ressources-fisa/tp08/". La bibliothèque est disponible à l'adresse : <https://github.com/siu/minunit>.

L'application est découpée en quatre fichiers dont le diagramme de composants est donné sur la figure -fig. 1 :

1. Un fichier qui contient les fonctions de calcul de la ration calorique (nommé "caloric_intake.c").
2. Un fichier contenant les déclarations des fonctions et des constantes du programme (nommé "caloric_intake.h").
3. Un fichier qui va permettre de tester individuellement chaque fonction du fichier "caloric_intake.c" avant de les intégrer dans le programme final (nommé "test_caloric_intake.c").
4. Un fichier comportant la fonction `main()` qui enchaîne les fonctions de calcul de la ration (nommé "main.c" dans l'exemple ci-dessous).

Les fichiers "caloric_intake.h" et "caloric_intake.c" vous sont fournis presque entièrement complets. Vous n'aurez donc à vous soucier, dans un premier temps, que du fichier de validation "test_caloric_intake.c", puis dans un second temps, de l'application finale avec le fichier "main.c".

**Figure 1:** Diagramme de composants

3 Travail à réaliser

1. Placez-vous dans votre dossier "odl" et créez le sous-dossier "tp08", puis placez-vous dans ce dernier. "tp08" sera votre dossier de travail pour cette séance et vous y effectuerez toutes vos actions.
2. Copiez l'ensemble des fichiers ".c" et ".h" du dossier "ressources-odl-fisa/tp08/" dans votre dossier de travail "tp08" en respectant la structure arborescente dans laquelle sont placés les fichiers.
3. Réalisez dans le fichier "test_caloric_intake.c" les tests de chacune des fonctions de manière à couvrir le maximum de cas. Pour réaliser ces tests vous pourrez vous aider du fichier "data.csv" qui contient plusieurs centaines de données et que les fichiers "util_csv.h" et "util_csv.c" vous permettront de manipuler.
4. Une fois les tests validés (et éventuellement les fonctions corrigées), vous pourrez compléter le fichier "main.c".
5. Proposez un fichier "Doxyfile" permettant de générer la documentation.

4 Livrable

Avant la date limite, déposez sur la plateforme Moodle l'archive compressée "rapport-tp08-prenom_nom.tgz" qui contiendra le contenu de tout le dossier "tp08", excepté les fichiers objets, les exécutables et la documentation générée par **Doxygen**.

Vous penserez à inclure dans le dossier "tp08" un fichier "README.md" qui :

- identifie l'auteur du TP ;
- décrit brièvement le contenu du dossier (objectif et composants) ;
- donne la ligne de commande pour compiler le programme de test ainsi que le programme final.

5 Résumé

Dans ce TP vous avez mis en oeuvre des tests unitaires et vous vous êtes familiarisés avec une compilation multifichiers sans l'utilisation d'un fichier "Makefile".

Références

- [1] J.D. Robertson et D.D. Reid. "Standards for the basal metabolism of normal people in Britain". Lancet, vol. 1, pp. 940-943, 1952.
- [2] D. Dubois et E.F. Dubois. "A formula to estimate the approximate surface area if height and weight be known". Arch Intern Med., vol. 17, pp. 863-871, 1916.
- [3] J.A. Harris et F.G. Benedict. "A Biometric Study of Human Basal Metabolism". Proceedings of the National Academy of Sciences of the United States of America, vol. 4(12), pp. 370-773, 1918.