

La méthode de développement descendante (top-down)

DAKKAR Borhen-eddine

Lycée le Corbusier

BTS SN-IR

Table des matières

- 1 Développement d'un programmes C
 - Etapes de développement d'un programmes C
- 2 Comment créer un programme en C?
 - Edition du programme
 - La compilation du programme
 - L'édition de liens
- 3 Cycle de vie du programme
- 4 Application
 - Exemple d'application

Table des matières

- 1 Développement d'un programmes C
 - Etapes de développement d'un programmes C
- 2 Comment créer un programme en C?
 - Edition du programme
 - La compilation du programme
 - L'édition de liens
- 3 Cycle de vie du programme
- 4 Application
 - Exemple d'application

Etapes de développement d'un programmes C

- Nous allons présenter la méthode de développement descendante (top-down) qui permet de convertir des problèmes de programmation en programmes C fonctionnels.
- L'approche descendante (également connue sous le nom de conception par étapes) est essentiellement la décomposition d'un système afin de mieux comprendre ses sous-systèmes de composition en mode d'ingénierie inverse.
- Dans une approche descendante une vue d'ensemble du système est formulé, en précisant, mais pas des détails, des sous - systèmes de premier niveau. Chaque sous-système est ensuite raffiné dans encore plus en détail, parfois plusieurs niveaux de sous-système supplémentaires, jusqu'à ce que l'ensemble du cahier des charges est réduit à la base des éléments.

Etapes de développement d'un programmes C

- La méthode top-down se compse de cinq étapes :
 - ➊ Déterminez les éléments de sortie souhaités que le programme doit produire.
 - ➋ Déterminez les éléments d'entrée.
 - ➌ Concevez le programme comme suit:
 - ➐ Sélectionnez un algorithme pour transformer les éléments d'entrée à la sortie désirée.
 - ➑ Vérifiez l'algorithme choisi à la main, en utilisant des valeurs d'entrée spécifiques.
 - ➒ Déterminer les noms de variables pour l'algorithme sélectionné.
 - ➍ Codez l'algorithme en C.
 - ➎ Testez le programme en utilisant les données de test sélectionnées.

Les étapes 1 et 2 de la procédure de développement sont appelées la **phase d'analyse**, l'étape 3 est appelée **phase de conception**, l'étape 4 la **phase de codage** et l'étape 5 la **phase de test**.

Etapes de développement d'un programmes C

- **La phase d'analyse (étapes 1 et 2) :** nous nous intéressons à l'extraction de toutes les informations d'entrée et de sortie fournies par le problème. Considérons le problème de programmation suivant :

La résistance électrique d'un fil métallique, en ohms, est donnée par la formule $r = (ml)/a$, où m est la résistivité du métal en ohms-mètres; l est la longueur du fil en mètres; et a est la section du fil en mètres carrés.

En utilisant ces informations, écrivez un programme C pour calculer la résistance d'un fil en cuivre qui mesure 20 mètres de long, avec une section de 2,5 mm² ($2,5 \times 10^{-6} m^2$) mètres carrés. La résistance du cuivre est de 17.10^{-8} ohms-mètres.

Etapes de développement d'un programmes C

- **Étape 1: Déterminez la sortie souhaitée** Pour déterminer la sortie du programme, l'énoncé du problème utilisera des mots tels que calculer, imprimer, déterminer, trouver ou comparer. Ces mots là peuvent être utilisés pour déterminer les sorties souhaitées.

Pour notre exemple, la phrase clé est « pour calculer la résistance d'un fil en cuivre ». Cela identifie clairement un élément de sortie.

- **Étape 2: déterminer les entrées du programme** cette étape de processus du développement exige que nous identifions tous les éléments d'entrée. Il est essentiel de faire la distinction entre les éléments d'entrée et les valeurs d'entrée.

Ici, les éléments d'entrée sont la résistivité, m , la longueur du fil, l , et la section du fil, a .

Etapes de développement d'un programmes C

- **Étape 3a: Sélectionnez un algorithme** : d'après l'énoncé du problème, il est clair que l'algorithme de transformation des éléments d'entrée à la sortie souhaitée est donné par la formule $r = (ml)/a$.

Il faut rappeler que que l'algorithme représente une description de la façon dont les entrées seront transformées en sorties. Alors le psedocode de l'algorithme est donné comme suit :

Début

Réel : m, l, n;

$r := (ml)/a$

Ecrire ("La résistance est égale à", r, "ohms")

FIN

Etapes de développement d'un programmes C

- **Étape 3b: Faites un calcul manuel** effectuer un calcul manuel nécessite que nous ayons des valeurs d'entrée spécifiques qui peuvent être appliqué à l'algorithme. Les trois valeurs d'entrée sont : la longueur du fil est de 20 mètres, la section est de 2,5 mm² ($2,5 \times 10^{-6} m^2$) mètres carrés, et la résistance du cuivre est de $17 \cdot 10^{-8}$ ohms-mètres.
En remplaçant ces valeurs dans la formule, nous obtiendrons la résistance de 0.136 ohms.
- **Étape 3c: Sélectionnez les noms de variables** la dernière étape de la phase de conception consiste à choisir les noms des variables. Utilisons les variables nommées **restv**, **sect**, et **longueur** pour la résistivité, la section et la longueur des éléments d'entrée, respectivement; et une variable nommée **resist** pour la sortie calculée (noms symboliques).

Etapes de développement d'un programmes C

- **Étape 4 : Codez l'algorithme en C**

```
#include <stdio.h>
/* Clcul de la résistance d'un fil */
main( )
{
    double restv, sect, longueur, resist;
    restv = 1.7e-8;
    sect = 2.5e-6;
    longueur = 125;
    resist = (restv*longueur)/sect;
    printf("\nLa resistance est egale a %lf", resist);
}
```

Étapes de développement d'un programmes C

- **Étape 5 : Testez le programme** Le but des tests est de vérifier qu'un programme fonctionne correctement et réellement répond à ses exigences. Une fois les tests terminés, le programme peut être utilisé pour calculer les sorties avec différentes données d'entrées sans avoir besoin de retester.

Etapes de développement d'un programmes C

- Le tableau suivant montre les phases de développement et la quantité relative d'effort consacrée à chaque phase dans les projets de programmation :

La phase	Étape de développement	Effort
La phase d'analyse	Etapes 1 et 2	10%
La phase de conception	Etape 3	20%
La phase de codage	Etape 4	20%
La phase de test	Etape 5	50%

Table des matières

- 1 Développement d'un programmes C
 - Etapes de développement d'un programmes C
- 2 Comment créer un programme en C?
 - Edition du programme
 - La compilation du programme
 - L'édition de liens
- 3 Cycle de vie du programme
- 4 Application
 - Exemple d'application

- L'édition ou la saisie d'un programme consiste à créer, à partir d'un clavier, le texte d'un programme qu'on nomme « programme source ».
- Le fichier qui contient le text est nommé « fichier source ».
- En général, un fichier C est caractérisé par l'extension "c", par exemple **mon_fichier.c**. Les caractères derrière le point sont appelés **extension**.

La compilation du programme

- Elle consiste à traduire le programme source en langage machine, en faisant appel à un programme nommé compilateur.
- Tout d'abord, il y a le traitement par le préprocesseur qui consiste à inclure un segment de code source disponible dans un autre fichier (par exemple les appels commençant par le caractère #).
- Le préprocesseur permet donc de remplacer des symboles d'un fichier source par «autre chose», c'est-à-dire par une chaîne de caractères quelconque que le compilateur traitera en lieu et place du symbole.
- La compilation représente la traduction en langage machine du texte en langage C fourni par le préprocesseur.

- Le module objet créé par le compilateur n'est pas directement exécutable. Il lui manque les différents modules objet correspondant aux fonctions prédéfinies (comme printf, scanf, sqrt).
- L'éditeur de liens va aller chercher dans la bibliothèque les modules objet nécessaires pour ces fonctions.
- Après l'édition de liens, un fichier sera généré. Ce fichier est appelé fichier exécutable.
- Un fichier exécutable est un ensemble d'instructions en langage machine, qui pourra être exécuté sans faire appel à l'environnement de programmation en C.

Table des matières

- 1 Développement d'un programmes C
 - Etapes de développement d'un programmes C
- 2 Comment créer un programme en C?
 - Edition du programme
 - La compilation du programme
 - L'édition de liens
- 3 Cycle de vie du programme
- 4 Application
 - Exemple d'application

Cycle de vie du programme

- Le cycle de vie d'un programme est divisé en trois étapes principales, l'élaboration de programmes, la documentation du programme et la maintenance du programme.
- **La phase de développement** : est l'endroit où un programme est initialement développé. C'est à cette étape où les exigences doivent être comprises et la structure du programme est planifié selon la procédure de développement descendante (top_down).
- **L'étape de la documentation** : elle consiste à créer à la fois dans le programme et dans des documents séparés un support suffisant pour les utilisateurs et les programmeurs. Il sera utilisé comme référence pour le programme.
- **La phase de maintenance** : le programme peut être modifié ou amélioré au fur et à mesure que de nouvelles demandes et exigences sont obtenues ou des erreurs de programme sont détectées.

Table des matières

- 1 Développement d'un programmes C
 - Etapes de développement d'un programmes C
- 2 Comment créer un programme en C?
 - Edition du programme
 - La compilation du programme
 - L'édition de liens
- 3 Cycle de vie du programme
- 4 Application
 - Exemple d'application

Exemple

L'oxygène utilisé dans le processus de combustion du charbon se combine avec le carbone et le soufre du charbon pour produire à la fois du dioxyde de carbone et du dioxyde de soufre.

Le dioxyde de soufre se combine avec l'eau et l'oxygène de l'air pour former de l'acide sulfurique, lui-même transformé en ions hydronium séparés et en des sulfates. Les ions hydronium tombent sur la terre soit comme composants de la pluie ou comme dépôt sec et changent le niveau d'acidité de lacs et de forêts. Le niveau d'acide des pluies et des lacs est mesuré sur une échelle de pH en utilisant la formule :

$$pH = -\text{Log}_{10}(\text{concentration d'ions hydronium})$$

où la concentration d'ions hydronium est mesurée en moles/litre. Une valeur de pH de 7 indique une valeur neutre (ni acide ni alcaline), tandis que les niveaux au-dessous de 7 indiquent la présence d'un acide et des niveaux supérieurs à 7 indiquent la présence d'une substance alcaline.

Exemple

En utilisant la formule du pH, nous écrivons un programme C qui calcule le niveau de pH d'une substance basé sur une valeur entrée par l'utilisateur pour la concentration d'ions hydronium.

Nous utiliserons la procédure de développement descendante (top-down) décrite précédemment:

- **Étape 1: Déterminez la sortie souhaitée** Bien que l'énoncé du problème fournit des informations techniques sur la composition des pluies acides, du point de vue de la programmation c'est un problème assez simple. Ici, il n'y en a qu'une seule sortie requise qui est le **niveau de pH**.
- **Étape 2: déterminer les entrées du programme** Pour ce problème, il n'y a qu'une seule entrée, le niveau de **concentration des ions hydronium**.

Exemple

- **Étape 3a: Sélectionnez un algorithme** Le traitement nécessaire pour transformer l'entrée à la sortie requise est une utilisation assez simple de la formule de pH fournie.

Algorithme 3 : Calcul de pH

Début

Réel : C;

Ecrire ("Entrer la valeur de la concentration d'ions hydronium")

Lire (C)

$\text{pH} := -\text{Log}_{10}(C)$

Ecrire ("La valeur du pH = ", pH)

FIN

- **Étape 3b: Faites un calcul manuel** Dans cette étape nous allons effectuer un calcul manuel. Le résultat de ce calcul peut puis être utilisé pour vérifier le résultat produit par le programme. Supposant que nous avons une concentration d'hydroniurn de 0.0001, le niveau de pH est calculé comme suit: $-\text{Log}_{10}(10^{-4}) = 4.$

Exemple

- **Étape 3c : Sélectionnez les noms de variables** La dernière étape requise avant que l'algorithme sélectionné ne soit décrit en C est de sélectionner la variable nommée pour l'entrée et la sortie, et toutes les variables intermédiaires requises par l'algorithme. Pour ce problème, les noms des variables seront : **hydron** pour la variable d'entrée et le nom **niveau_ph** pour le niveau de pH calculé.
- **Étape 4 : Codez l'algorithme en C**

```
#include <stdio.h>
#include <math.h>
main( )
{
    float hydron, niveau_ph;
    printf("Entrer la concentration d'ions hydronium : ");
    scanf ("%f", &hydron);
    niveau_ph = -log10(hydron);
    printf("\n Le niveau de pH est %f", niveau_ph);
}
```

Exemple

- **Étape 5 : Testez le programme** Nous pouvons tester notre programme en lui fournissant la valeur de concentration d'ions hydronium par exemple 0.0001.

