

La programmation

I- Position dans le cycle de vie :

La phase de codage suit immédiatement la phase de conception. Elle consiste à produire le logiciel et sa documentation.

A partir d'une hiérarchie de module et de sous module, le développement descendant consiste à commencer par la racine de la hiérarchie et de se diriger vers les modules en bas. Le développement ascendant est donc le processus inverse.

II- Facteurs influençant l'activité de programmation :

1. Le langage de programmation et la créativité du programmeur :

L'activité de programmation s'appuie sur la créativité du programmeur et des possibilités offertes par le langage.

a- choix des noms :

Influence la lisibilité du programme. La nature et la fonction des objets et les unités constituant un modèle doivent être refléter dans leur identificateur.

Le langage de programmation limite la longueur d'un identificateur donné, ce qui amène le programmeur à choisir une technique pour affecter des identificateurs lisibles et significatif. Quelque soit la technique choisie, elle doit être respectée dans tout le code.

Exemple :

Si un programme calcule les orbites des satellites, il doit manipuler les grandeurs suivantes : masse du satellite, vitesse du satellite et son accélération et la masse terrestre.

1^{ère} technique : Identificateur explicite :

Masse satellite, vitesse satellite, accélération satellite, masse terre.

2^{ème} technique : Identificateurs abrégés :

Vit-Sat, Acc-Sat, Mas-Sat, Mas-Ter.

b- Agencement du programme :

Affecte la lisibilité du programme, la clarté du code produit et mesuré selon différents critères :

- L'espacement entre les différentes parties d'un module si nécessaire (entre parties déclarations et le reste du code par exemple).
- La mise en œuvre des mots réservés par une typographie particulière (couleurs différentes, ...).
- La mise en page cohérente (intention pour bloc imbriqué).

c- Documentation :

L'emploi des commentaires pour mettre en évidence la signification des objets et unités d'un module est une bonne stratégie de documentation du code une fois le langage sous-jacent dispose d'une syntaxe de commentaire.

d- Allure du flot de contrôle :

Affecte la fiabilité du programme. Il faut toujours s'efforcer de rendre le flot de contrôle linéaire de façon que l'exécution du programme décent (soit correct) le long du programme.

e- Quelques règles :

- Eviter le minimum possible d'employer l'instruction GO TO dans un langage structuré. En revanche, en cas de survenue d'exception, l'emploi de cette instruction améliore l'acheminement du programme (décider de ne plus continuer l'exécution en signalant la cause avant de quitter).

- L'emploi de l'instruction IF...THEN...ELSE avec une profondeur exagérée, amène à des programmes difficiles à comprendre. Il vaut mieux de trouver d'autres alternatives plus adaptées s'ils sont offertes par le langage choisi tel que les structures multi choix(switch, condition combinées avec des opérateurs logiques).

2. Les outils de développement :

La mise en pratique d'outils qui automatisent une partie du travail du programmeur augmente de façon importante la productivité de ce dernier.

L'expérience a prouvé que si les outils de développement sont accessibles à partir d'un système global (IDE : Environnement de développement intégré (visual basic,...), ça facilite le déroulement des tâches du programmeur.

Exemple d'outils :

- Editeur de code (mise en page, include,...).
- Compilateur.
- Débogueur.

3. Machine/système d'exploitation :

Affecte la portabilité d'un programme. Il faut réduire et isoler les parties d'un programme dépendant d'une machine ou d'un système d'exploitation.

a- Dépendances dues à la machine :

- **Longueur d'un mot machine :** Soit dans programme, une fonction faisant des calculs sur des entiers qui peuvent atteindre 500000. la machine de développement encode les entiers sur 32 bits.

- Le programme fonctionne correctement sur la machine de développement. Le fait de la porter sur une machine 16 bits provoquera des erreurs logiques (les valeurs entières vont être inférieure à 32767).

- Jeu de caractères : code ASCII, code EBCDIC,...

b- Dépendances dues au système d'exploitation :

Un langage de programmation dépend des caractéristiques du système d'exploitation qui l'accueille.

- **Les bibliothèques :** (Influence programmatique) : l'utilisation des fonctions de ces bibliothèques peut compliquer le partage d'un programme (bibliothèque sans interface standard, bibliothèque non disponible sur l'autre système,...).

- **Les fichiers :**

- ❖ Convention de nommage de fichiers peut être différents d'un système à l'autre.
- ❖ Nombre maximum de fichiers qu'un programme peut ouvrir simultanément.
- ❖ Les primitives d'accès : la disponibilité d'accès aléatoire n'est pas garantie sur tous les systèmes.

- ❖ Les E/S : taille du terminal.

4. Milieu de travail :

Chaque programmeur doit disposer d'un espace qui lui est propre où il peut se concentrer.

- ❖ La lumière naturelle et l'environnement extérieur doivent être présents.
- ❖ Chaque programmeur doit pouvoir réorganiser son milieu de travail de manière personnalisable.

5. Les langages de programmation :

a- Les langages d'assemblage :

Il existe une correspondance univoque entre une instruction machine et une instruction de langage d'assemblage. Le programmeur doit transcrire les actions à réaliser en séquence d'instructions pour la machine.

Tout langage d'assemblage est lié à la famille d'ordinateur pour lequel il a été développé.

Inconvénients :

- langage difficile.
- Forte portabilité d'erreurs.

b- Les langages de réalisation de systèmes :

- Ils sont nés pour résoudre des difficultés liées à la programmation en assembleur.
- Ils offrent la possibilité de la programmation système et l'accès au niveau bas de la machine.
- Ils augmentent le degré de portabilité du code.

C est un exemple (réalisation de 90% d'Unix).

- Ils offrent des structures de contrôles évoluées et le typage de données.

c- Les langages statiques de haut niveau :

- Offrent des structures de haut niveau et des déclarations de variables.
- Ils ne laissent aucun contrôle au programmeur on ce qui concerne les instructions machines générées par le compilateur.
- Seulement allocation statique de la mémoire est disponible.

Exemple : FORTRAN, COBOL, C.

d- Les langages à haut niveau à structure de bloc :

- En plus des possibilités offertes par les langages statiques, ils ont introduit une forme limitée d'allocation dynamique de la mémoire (structure de bloc).

Exemple : PASCAL, ALGOL, C, C++.

e- Les langages dynamiques de haut niveau :

La mémoire est complètement gérée dynamiquement. Ils sont utilisés en intelligence artificielle. Exemple : PROLOG, LISP.