

Le Langage C

Le C ANSI

- C créée en 1972 par Denis Ritchie pour développer UNIX.
- Définition rigoureuse en 1978 (Kernighan & Ritchie).
- Normalisation ANSI (American National Standard Institut) en 1988.
- Puis ISO (International Standardisation Organization) et Europe.

Les atouts du C

- L'un des langages les plus utilisés
- Instructions et structures de haut niveau
- un code rapide grâce à
 - un compilateur performant.
 - Des instructions proches du langage machine

Objectifs de la normalisation

- Améliorer et sécuriser le langage.
- Assurer l'ascendance
(d'anciens programmes peuvent encore tourner).
- Exemple de problèmes:
déclaration de fonctions

Créer un premier programme

1. Écrire le code source

- Éditeur de texte
- Décomposer le problème
- Définir les échanges

2. Créer un exécutable

- Précompilation
- Code objet
- Édition de lien

Exemple d'un programme en C

```
/* Exemple de programme en C */ . . .
```

Commentaire

```
#include <stdio.h>
```

```
#define ICI 16
```

Directives au
préprocesseur

```
main()
```

```
{
```

Begin

```
int bureau; . . .
```

Déclaration de
variable

```
bureau = ICI; . . .
```

Assignation

```
printf("J'aime le C (au lieu de pascal).\n"); . . .
```

```
printf("Mon bureau est au ", %d, " en Dma.\n");
```

```
Return 0;
```

Fonction
printf()

Bloc

End

Créer un premier programme

- Fonctions et variables : instructions à effectuer, valeurs à manipuler
- Séparateurs : espace, points-virgules
- Délimiteurs de bloc : accolades

Commentaire	/* Mon premier programme */	
Directive d'inclusion	#include <stdio.h>	
	main()	
	{	
Déclaration des variables	int i = 12;	
	int j, k;	
	double x;	
Instructions	k = i+2;	
	printf("Hello world !\n");	
	}	Fonction main

Premier programme

```
/*-----
```

```
tva.c – 24/03/2009- Tairi–  
Calcul et affiche un prix ttc
```

```
-----*/
```

```
#include <stdio.h>
```

```
#define TVA 18.6
```

```
Int main(void){
```

```
    float ht,ttc;
```

```
    puts(“entrez le prix ht.”);
```

```
    scanf(“%f”,&ht);
```

```
    ttc=ht*(1+(TVA/100));
```

```
    printf(“prix TTC: %f\n”,ttc);
```

```
    return 0;
```

```
}
```

- Commentaires

- Directives préprocesseur (#)

- Inclure un fichier, définir une constante,...
- Traitement de texte

- Déclarations de variable

Type liste_var_de_ce_type;

- Affectations de variables(=)

- Manipulations

- Fonctions de la librairie standard
- Expressions

- IMPORTANT:

Tout objet (fonction, variable) doit être déclaré avant d'être utilisé!

Dans une définition de fonction

- Liste d'arguments entre parenthèses
- Liste d'instructions entre accolades
- Une valeur de retour typée

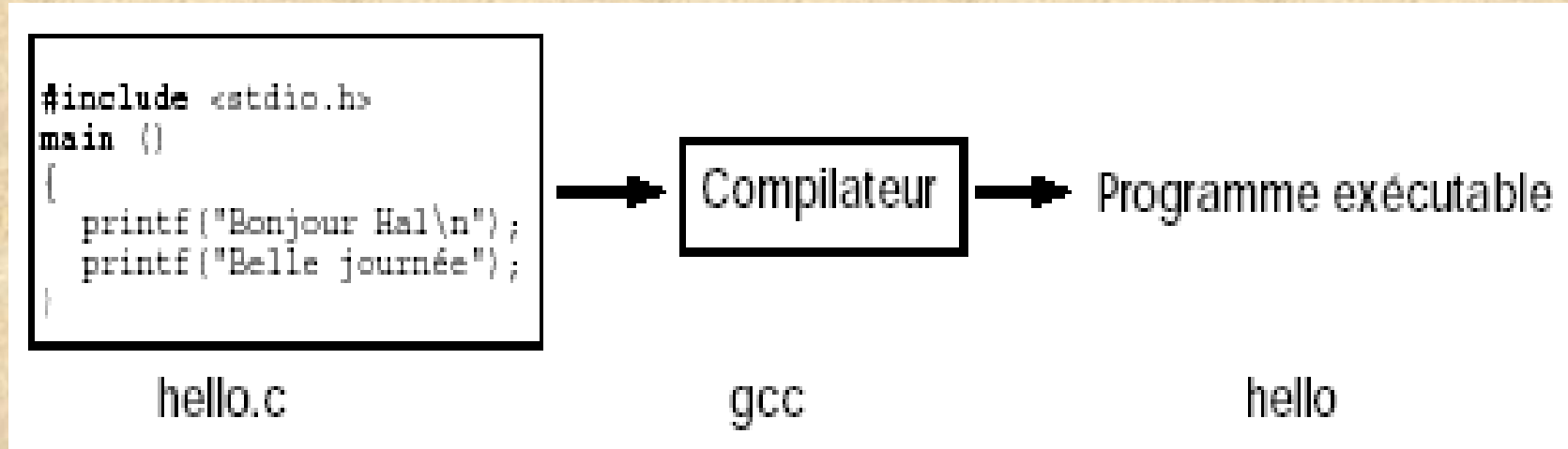
fonction **main**

- La fonction **main** est la fonction principale des programmes en C: Elle se trouve obligatoirement dans tous les programmes. L'exécution d'un programme entraîne automatiquement l'appel de la fonction **main**.

- **main()**
 { | <déclarations>
 | <instructions>
 | return 0;
 }

return 0; qui indique à l'environnement que le programme s'est terminé avec succès, sans anomalies ou erreurs fatales.

Création d'un exécutable



Trois étapes assurées par gcc :(Gnu C Compiler)

1. précompilation
2. Création du code objet
3. Édition de lien

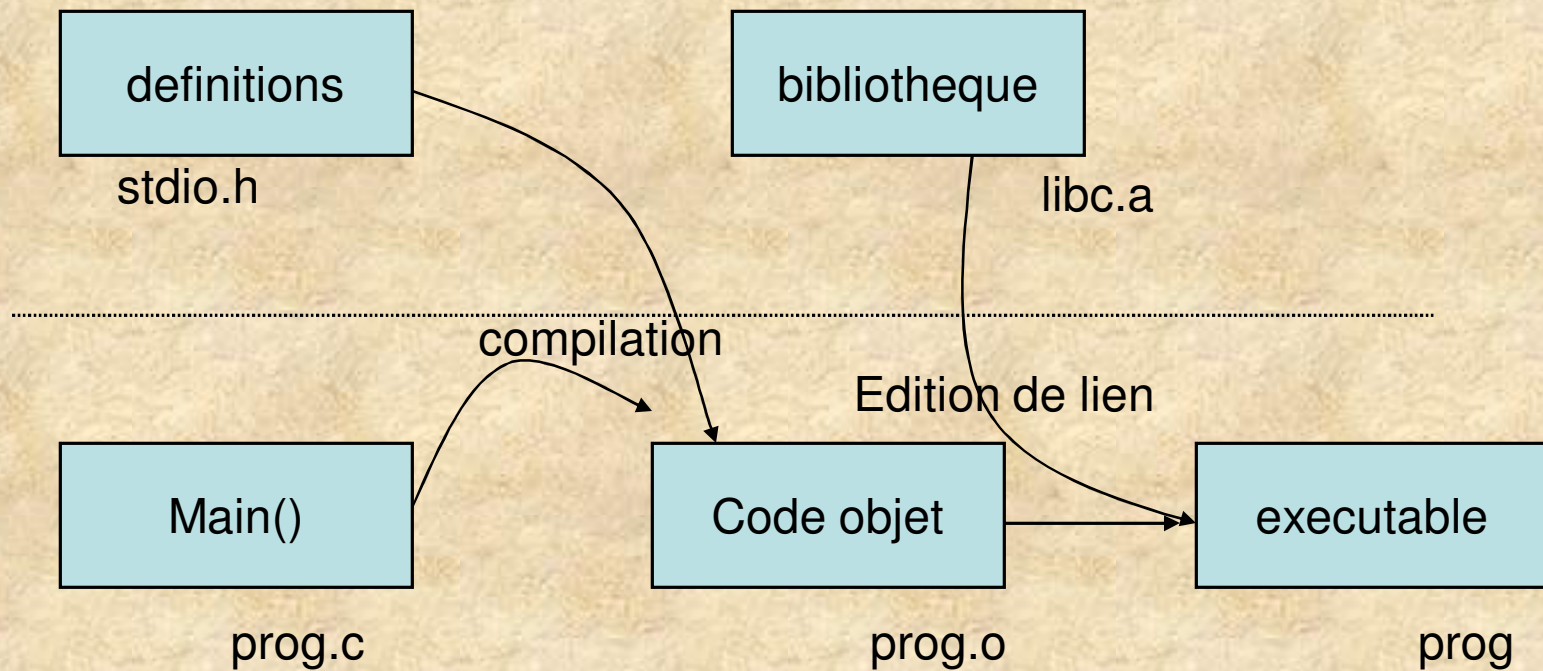
Module objet (fichier.o)

- Le compilateur s'assure que:
 - La syntaxe est correcte
 - Les variables sont bien déclarées
 - Les fonctions bien utilisées
- Ne s'assure pas que:
 - La mémoire est bien gérée
- Convertit en langage machine

Edition de lien

- Le fichier objet n'est pas exécutable:
 - Ne contient pas de main
 - Dépend de fonctions ou variables externes
 - ...
- L'édition de lien combine les fichiers objets contenant les réalisations de tout les modules pour former l'exécutable.

Edition de lien (standard)



Les erreurs

- Erreurs de compilation
 - Erreur de syntaxe, déclaration manquante, parenthèse manquante,...
- Erreur de liens
 - appel a des fonctions dont les bibliothèques sont manquantes
- Erreur d'exécution
 - Segmentation fault, division par zéro
- Erreur logique