

Projet Génie Logiciel

# Compilateur Decac

## Manuel utilisateur

---

**Équipe GL2**

**Étudiants : Élina Houdé, Julien Pinto Da Fonseca,  
Léa Solo Kwan, Yin Xu**

# Sommaire

<b>Utilisation standard du compilateur Decac</b>	<b>3</b>
<b>1. Options du compilateur Decac</b>	<b>3</b>
1.1. Options disponibles	4
1.2. Options disponibles prochainement	4
<b>2. Messages d'erreurs</b>	<b>5</b>
2.1. Messages de type DecacFatalError	5
2.2. Messages de type DecacCompilerError	6
2.3. Messages de type ContextualError	7
<b>3. Limitations du compilateur</b>	<b>9</b>

---

## Utilisation standard du compilateur Decac

Le compilateur Decac est un compilateur en ligne de commande utilisable sous les distributions GNU/Linux. Il permet notamment de générer un fichier assembleur *.ass* à partir d'un programme deca (exemple : *hello.deca*). Le fichier assembleur généré est placé dans le même dossier que le fichier d'entrée.

L'appel au compilateur se fait via la commande *decac* dont l'utilisation standard est détaillé ci-dessous :

***decac <fichiers deca>...***

La commande *decac* prend en paramètre un ou plusieurs programmes deca : il s'agit de la liste des fichiers *.deca* à compiler. Le nom de chaque fichier peut être précédé du chemin permettant d'y accéder, i.e. que chaque fichier peut être indiqué sous la forme *<chemin/nom\_fichier.deca>*.

### 1. Options du compilateur Decac

Le compilateur Decac peut être exécuté avec diverses options :

***decac [[-p | -v] [-n] [-r X] [-d]\* [-P] [-w] <fichiers deca>...] | [-b]***

*Syntaxe d'utilisation*

La commande *decac*, sans argument, affiche la liste de toutes les options, implémentées ou non.

### 1.1. Options disponibles

- ***b (banner)*** : affiche une bannière indiquant le nom de l'équipe.
- ***p (parse)*** : affiche la décompilation de l'arbre construit lors de la vérification syntaxique du fichier .deca. Ne génère pas de fichier assembleur.
- ***v (verification)*** : vérifie le code contenu dans le fichier .deca sans le compiler. Ne retourne rien en l'absence d'erreur.

N.B. Les options '*p*' et '*v*' sont incompatibles.

### 1.2. Options disponibles prochainement

- ***n (no check)*** : supprime les tests de débordement à l'exécution (i.e. débordement arithmétique, débordement mémoire, déréférencement de null).
- ***r X (registers)*** : limite les registres banalisés disponibles à R0 ... R{X-1}, avec  $4 \leq X \leq 16$ .
- ***d (debug)*** : active les traces de debug. Répéter l'option plusieurs fois augmente le niveau de debug.
- ***P (parallel)*** : s'il y a plusieurs fichiers sources, lance la compilation des fichiers en parallèle pour accélérer la compilation.

## 2. Messages d'erreurs

Ci-dessous sont détaillés les différents messages d'erreurs pouvant être retournés par le compilateur Decac. Ces erreurs sont répartis en trois catégories bien distinctes :

- **DecacFatalError** : survient lors de l'utilisation d'un objet inexistant pendant l'étape de génération du code assembleur. Il s'agit d'une erreur rare et généralement critique touchant au code source du compilateur Decac ;
- **DecacCompilerError** : survient lors d'une mauvaise exécution du compilateur Decac. Il s'agit généralement d'une commande incorrecte (options incompatibles, mauvais fichier d'entrée...);
- **ContextualError** : peut survenir lors de l'analyse contextuelle d'un code source si un problème est rencontré dans celui-ci.

### 2.1. Messages de type DecacFatalError

- **"Numéro de registre inexistant : ..."**

Cette erreur peut apparaître lors de l'utilisation de la classe *RegisterManager*. Elle signale que vous essayez d'allouer ou de libérer un registre qui n'existe pas ou un registre scratch (registres R0 et R1).

N.B : Les registres scratch n'ont pas besoin d'être réservés.

- **"Un type de label inexistant a été utilisé."**

Cette erreur peut apparaître lors de l'utilisation de la classe *StackManager*. Elle signifie que le type de label que vous essayez d'utiliser n'existe pas. Elle peut survenir lorsque le type de label utilisé est nul ou qu'une mauvaise manipulation a été effectuée dans le code source du compilateur.

## 2.2. Messages de type DecacCompilerError

- **"L'entrée suivante n'est pas reconnue : ..."**

Ce message d'erreur signifie que la commande pour exécuter le compilateur Decac n'est pas correcte.

- **"L'option suivante n'est pas reconnue : ..."**

Ce message d'erreur signifie que l'option spécifiée n'existe pas. Celle-ci est peut-être mal orthographiée.

- **"Le fichier suivant n'existe pas : ..."**

Ce message d'erreur signifie que le compilateur ne trouve pas le programme deca spécifié en paramètre. Le nom du fichier ou le chemin d'accès est incorrect ou alors il ne s'agit pas d'un programme deca.

- **"Option -r X incorrecte ! Contrainte à respecter :  $4 \leq X \leq 16$ ."**

Ce message d'erreur signifie que le paramètre X spécifié pour l'option -r est soit trop petit (strictement inférieur à 4), soit trop grand (strictement supérieur à 16).

- **"Les options -p et -v sont incompatibles."**

Ce message d'erreur signifie que vous avez tenté d'utiliser les options -p et -v simultanément. Or, il s'agit d'un comportement interdit : veuillez utiliser ces options séparément.

- **"L'option -b ne peut pas être utilisée avec une autre option et ne prend pas de paramètre."**

Ce message d'erreur signifie que vous avez essayé d'utiliser l'option -b avec une autre option ou un fichier source. Or, ces deux comportements sont interdits : la seule utilisation correcte de l'option -b est la suivante :

<i>decac -b</i>
-----------------

### 2.3. Messages de type `ContextualError`

- **"La variable suivante est de type void : ..."**

Ce message d'erreur signifie que vous avez essayé de déclarer une variable de type void. Or, cela ne respecte pas les spécifications du langage Deca.

- **"La variable suivante a déjà été déclarée précédemment : ..."**

Ce message d'erreur signifie que vous avez essayé de déclarer une variable qui a déjà été déclarée auparavant. Cette erreur survient lors de l'utilisation de noms de variables qui ne sont pas uniques.

- **"La variable suivante est indéfinie : ..."**

Ce message d'erreur signifie que la variable indiquée n'a pas été déclarée. Toute variable doit être déclarée avant de pouvoir être utilisée.

- **"Le type de variable suivant est indéfini : ..."**

Ce message d'erreur signifie que vous avez essayé d'utiliser un type de variable incorrect. En dehors des types *boolean*, *void*, *string*, *int* et *float*, tout type de variable doit être défini avant de pouvoir être utilisé.

- **"Le type de variable suivant est incompatible pour l'affectation : ..."**

Ce message d'erreur signifie que vous avez essayé d'affecter un type de valeur incompatible avec le type de la variable indiquée. L'affectation ne respecte pas une des deux conditions suivantes :

- Si la variable est de type *float*, la valeur à affecter doit être de type *float* ou de type *int* ;
- Si la valeur est de type *T1* et la variable de type *T2*, le type *T1* doit être un sous-type de *T2*.

- **"Deux expressions ne peuvent pas être comparées si leurs types sont les suivants : ..."**

Ce message d'erreur signifie que vous avez essayé de comparer deux expressions dont les types sont incompatibles pour la comparaison. Les types en question et les variables concernées sont indiquées à la fin du message.

- **"L'expression dans la condition booléenne doit être de type booléen. Or, elle est de type : ..."**

Ce message d'erreur signifie que vous avez essayé d'utiliser une expression non-booléenne en tant que condition. Ce comportement est interdit par les spécifications du langage Deca.



### 3. Limitations du compilateur

Le compilateur Decac ne prend actuellement pas en charge toutes les portions du langage Deca, notamment :

- le cast des variables ;
- l'opérateur *instanceof* ;
- l'opérateur *moins unaire* ;
- les variables de type booléen ;
- la lecture de la saisie de l'utilisateur via `readFloat()` ;
- les opérations arithmétiques ;
- les fonctions ;
- la programmation orientée objet.

De plus, le compilateur ne peut actuellement pas compiler de programme nécessitant l'utilisation de plus de 14 registres simultanément. En effet, le compilateur ne gère pas encore l'utilisation de la pile afin de stocker temporairement les valeurs des registres.