

Projet Génie Logiciel

Compilateur Decac

Manuel utilisateur

Équipe GL2

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

Sommaire

Utilisation standard du compilateur Decac	3
1. Options du compilateur Decac	3
1.1. Options disponibles	4
1.2. Options disponibles prochainement	4
2. Messages d'erreurs	5
2.1. Messages de type DecacCompilerError	5
2.2. Messages de type ContextualError	7
2.3. Messages de type CodegenError	15
3. Limitations du compilateur	17

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.
- ***P (parallel)*** : s'il y a plusieurs fichiers sources, lance la compilation des fichiers en parallèle pour accélérer la compilation.
- ***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.

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, dérérérencement de null).

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 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 essayez 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>

2.2. 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.

- **" L'opérateur moins unaire (symbol "-") doit s'appliquer sur une expression de type int ou float. Or, il est appliqué sur une expression de type : ..."**

Ce message d'erreur signifie que vous avez essayé d'utiliser l'opérateur moins unaire ("-") sur une expression qui n'est pas de type int ou float. Ce comportement est interdit par les spécifications du langage Deca.

- **" Deux expressions ne peuvent pas être additionnées, soustraites, divisées ou multipliées si elles ne sont pas de type int ou float. Or leurs types sont les suivants : ..."**

Ce message d'erreur signifie que vous avez essayé d'additionner, soustraire, diviser ou multiplier deux expressions qui ne sont pas de type int ou float. Ce comportement est interdit par les spécifications du langage Deca.

- **" L'opérateur % n'est utilisable qu'avec deux expressions de type int. Or il est actuellement utilisé avec des expressions de type : ..."**

Ce message d'erreur signifie que vous avez essayé d'utiliser l'opérateur % deux expressions qui ne sont pas de type int. Ce comportement est interdit par les spécifications du langage Deca.

- **" La fonction print n'affiche que des expressions de type int, float et string. Or vous essayez d'afficher une expression de type : ..."**

Ce message d'erreur signifie que vous avez essayé d'afficher des expressions qui ne sont pas de type int, float ou string. Ce comportement est interdit par les spécifications du langage Deca.

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

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

- **" La classe suivante n'est pas définie : ..."**

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

- **" L'opérateur new instancie une variable de type classe. Or, vous essayez d'instancier une variable de type : ..."**

Ce message d'erreur signifie que vous avez essayé d'instancier avec l'opérateur new une variable qui n'est pas de type classe. Ce comportement est interdit par les spécifications du langage Deca.

- **" L'attribut suivant a déjà été déclaré précédemment dans la classe : ..."**

Ce message d'erreur signifie que vous avez essayé de déclarer un attribut qui a déjà été déclaré auparavant dans la classe suivante. Cette erreur survient lors de l'utilisation de noms d'attributs qui ne sont pas uniques.

- **" Le type de l'attribut suivant est void : ..."**

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

- **" Le type du paramètre suivant est void : ..."**

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

- **" Vous essayez de redéfinir la méthode suivante avec une signature différente : ..."**

Vous essayez de redéfinir une méthode héritée. Cependant, la liste des paramètres pris en entrée de la nouvelle méthode ne correspond pas à la liste des paramètres de la méthode héritée au niveau du typage.

- **" La méthode suivante possède le même nom qu'un des attributs hérités : ..."**

Vous essayez de définir une méthode de nom X. Cependant, un des attributs de la classe supérieure utilise déjà ce nom.

- **" L'attribut suivant possède le même nom qu'une des méthodes héritées : ..."**

Vous essayez de définir un attribut de nom X. Cependant, une des méthodes de la classe supérieure utilise déjà ce nom.

- **" Vous essayez de redéfinir la méthode suivante avec un type de retour différent : ..."**

Vous essayez de redéfinir une méthode héritée. Cependant, le type de retour de la nouvelle méthode ne correspond pas à celui de la méthode héritée.

- **" Une méthode prend des paramètres avec un nom unique. Or, le paramètre suivant existe déjà : ..."**

Ce message d'erreur signifie que vous avez une méthode qui possède plusieurs fois le nom de paramètre suivant. Or une méthode ne prend que des paramètres avec un nom unique.

- **" Le mot-clef this ne peut être utilisé qu'au sein d'une classe. Or, vous l'utilisez dans la fonction main."**

Ce message d'erreur signifie que vous avez essayé d'utiliser le mot-clef this dans le fonction main. Or il ne peut être utilisé qu'au sein d'une classe.

- **" Vous essayez d'accéder à un attribut protégé. Or, le type X n'est pas un sous-type de Y."**

Cette erreur indique qu'il n'est pas possible d'accéder à l'attribut de type protégé. Pour accéder à un attribut protégé :

1. le type de l'expression doit être un sous-type de la classe courante ;
-

2. le type de la classe courante doit être un sous-type de la classe où le champ protégé est déclaré.

- **" L'accès aux champs d'une classe n'est possible que si le type de l'expression correspond à une classe. Or, l'expression X est de type Y. "**

Cette erreur indique que le type de l'expression X n'est pas de type classe donc que nous ne pouvons pas accéder au champ de la classe.

- **" L'accès aux méthodes d'une classe n'est possible que si le type de l'expression correspond à une classe. Or, l'expression X est de type Y. "**

Cette erreur indique que le type de l'expression X n'est pas de type classe donc que nous ne pouvons pas accéder aux méthode de la classe.

- **" Vous essayez d'accéder à la méthode suivante sans préciser la classe où elle est définie : "**

Cette erreur indique que vous appelez une méthode sans préciser la classe où elle est définie. Si la méthode `getX()` appartient à la classe A et que *a* est une instance A. Vous devez faire `a.getX()` pour accéder à la méthode et non juste `getX()`.

- **" La méthode est de type void : elle ne nécessite donc pas d'instruction "return" "**

Cette erreur indique que vous essayez d'effectuer un "return" dans une méthode de type void. Ce comportement est interdit par les spécifications du langage Deca.

- **" L'instruction return renvoie une expression dont le type n'est pas un sous-type du type de retour de la méthode : "**

Cette erreur indique que le type de l'expression renvoyée par return est invalide. Il doit être un sous-type du type de retour de la méthode.

- **" La méthode suivant n'attend aucun paramètre en entrée : "**

Cette erreur vous indique que la méthode que vous essayez d'utiliser avec des paramètres n'en prend aucun.

- **" La méthode suivant prend en entrée "**

Cette erreur vous indique que vous n'avez pas donné le bon nombre de paramètre à la méthode utilisée. Et vous donne le bon nombre de paramètre que la méthode doit prendre.

- **" L'opérateur instanceof n'accepte comme première opérande que l'instance d'une classe ou l'élément null. Or l'expression suivant est de type "**

Cette erreur indique que la première opérande que vous passez à l'opérateur instanceof est invalide. La première opérande doit être une instance de classe ou null.

- **" L'opérateur instanceof n'accepte comme seconde opérande qu'un type de classe. Or, la seconde opérande est de type : "**

Cette erreur indique que la seconde opérande que vous passez à l'opérateur instanceof est invalide. La première opérande doit être type.

- **" Vous essayez de caster l'expression suivante qui est de type void : "**

Cette erreur indique que vous essayez de caster une variable de type void . Ce comportement est interdit par les spécifications du langage Deca.

- **" L'expression suivante ne peut pas être castée dans le type : "**

Cette erreur indique que l'expression que vous essayez de caster dans un type est incompatible.

2.3. Messages de type `CodegenError`

- **" Erreur : dépassement arithmétique sur les flottants ou division par zéro. "**

Cette erreur est provoquée par une des situations suivantes :

- une division entière ou un reste par 0 provoque une erreur ;
- un débordement sur une opération arithmétique sur des flottants ;
- une division flottante par 0.0.

- **" Saisie incorrecte : la saisie est soit incorrecte (nombre entier attendu), soit celle d'un nombre trop grand (dépassement). "**

Cette erreur signifie :

- soit que vous avez saisi un nombre trop grand : dans ce cas vous devez le diminuer ;
- soit que votre saisie est incorrecte syntaxiquement : dans ce cas, veuillez bien saisir un nombre entier.

- **" Saisie incorrecte : la saisie est soit incorrecte (nombre réel attendu), soit celle d'un nombre trop grand (dépassement). "**

Cette erreur signifie :

- soit que vous avez saisi un nombre trop grand : dans ce cas vous devez le diminuer ;
- soit que votre saisie est incorrecte syntaxiquement : dans ce cas, veuillez bien saisir un nombre réel.

- **" Erreur : allocation impossible, tas plein. "**

Cette erreur indique que le tas est plein. C'est à dire que on ne peut plus allouer d'espace dans le tas.

- **" Erreur : pile pleine. "**

Cette erreur indique que la pile est pleine. C'est à dire que on ne peut plus allouer d'espace dans la pile.

- **" Erreur ; déréférencement de null. "**

Cette erreur indique que vous essayez d'utiliser une variable de type null. Cette erreur peut survenir quand vous essayez d'accéder au méthode d'une classe non initialisée.

- **" Erreur : conversion int -> float impossible. "**

Cette erreur indique que la conversion int vers float a échoué. Vérifiez le format de vos variables, la variable n'est pas codable sur un flottant.

- **" Erreur : cast float -> int impossible. "**

Cette erreur indique que le cast float vers int a échoué. Vérifiez le format de vos variables, la variable n'est pas codable sur un entier.

3. Limitations du compilateur

Les limitations du compilateur Decac concernent l'extension Trigonométrique. Pour plus de précision, veuillez consulter la documentation de la bibliothèque standard.