Fichiers fournis

Jump to bottom

VBrazhnik a modifié cette page on Jan 3, 2019 · 1 révision

La page du projet Corewar contient des liens pour télécharger les fichiers suivants:

Archiver `vm_champs.tar`

Dans cette archive, vous pouvez trouver un fichier exécutable de machine virtuelle compilé `corewar`, un programme de traduction compilé `asm`, ainsi que de nombreux champions dans le formulaire `.s` et les `.cor` fichiers.

Traducteur `asm`

L'affectation du projet indique que le programme `asm` traduit le code du langage d'assemblage écrit dans le `.s` fichier en bytecode, qui doit être placé dans le fichier avec l'extension `.cor`.

Mais en fait, le programme fourni ne prête aucune attention aux extensions des fichiers entrants. Il recherche simplement le dernier point du nom et remplace tous les autres contenus par une extension `.cor`.

La question "Dans quelle mesure cette approche est-elle optimale?" reste ouvert.

Après tout, cela rend les situations suivantes possibles, qui, bien que n'étant pas des erreurs au sens habituel, violent néanmoins le flux de travail déclaré:

```
$ ./asm batman.cor
Writing output program to batman.cor
```

```
$ ./asm batman
Writing output program to .cor
```

```
$ ./asm dc.heroes/batman
Writing output program to dc.cor
```

De plus, le programme fourni ne limite en aucun cas le nombre de champions acceptés pour traitement.

Par conséquent, de nombreux fichiers avec des extensions très différentes peuvent être spécifiés en tant qu'arguments à la fois. Certes, seul le dernier d'entre eux sera traité:

```
$ ./asm ant-man.s iron_man.s batman.s
Writing output program to batman.cor
```

Tous les autres arguments sont `asm` ignorés:

```
$ ./asm --undefined-flag incorrect_file.s batman.s
Writing output program to batman.cor
```

Comment vivre avec?

Puisque ce comportement n'est pas décrit dans le texte de la tâche, dans votre propre implémentation, vous pouvez limiter le nombre d'arguments acceptés à un seul fichier, qui doit avoir l'extension garantie `.s`.

Cela rendra le travail `asm` plus transparent et éliminera les vulnérabilités ci-dessus dans le flux de travail.

> À propos, la machine virtuelle ne vérifie pas non plus les fichiers entrants pour l'extension `.cor`.
>
> Et bien que `corewar` ce comportement ne soit pas une source de vulnérabilités pour un programme , dans son implémentation, cette vérification doit quand même être ajoutée afin de maintenir un style de comportement uniforme pour tous les composants du système.

Des champions

Il y a aussi des problèmes avec les exemples de champions fournis.

Malheureusement, tous les fichiers n'ont pas été écrits correctement. Par conséquent, le programme `asm` ne pourra pas traduire certains d'entre eux en bytecode et donnera une erreur.

Comment vivre avec?

Dans une telle situation, il convient de garder à l'esprit qu'il `champs/examples` n'y aura aucun problème avec les fichiers du dossier , mais le reste devra être vérifié personnellement.

Fichiers `op.c` et `op.h`

Le fichier contient une `op.c` structure qui décrit chaque opération définie en langage d'assemblage.

Nous pouvons écraser les données fournies dans cette structure sous n'importe quelle forme qui nous convient et les placer dans le projet. Il n'y a aucune restriction dans ce cas.

Nous sommes également fournis avec un fichier d'en-tête `op.h`. Il contient d'importantes constantes de préprocesseur qui définissent les paramètres de la machine virtuelle, ainsi que la syntaxe de l'assembleur.

Ce fichier doit être traité avec plus de soin et inclus dans le projet tel quel. Faire uniquement les changements les plus nécessaires.

Puisque pour participer au projet **"Corewar Championship",** sa présence est l'une des conditions:

> Il sera exécuté sur notre propre machine virtuelle, donc la configuration sera celle que vous décrirez dans votre fichier `op.h` qui sera joint.

Les changements nécessaires, qui doivent être effectués sur ce fichier, sont de le mettre en conformité avec la norme. Étant donné que le fichier fourni n'est pas conforme aux règles de formatage établies, Norminette affichera une série d'avertissements concernant les erreurs détectées lors de sa vérification.

Si la participation à la compétition "Corewar Championship" n'est pas prévue, alors `op.h` ne vous inquiétez pas du sort du fichier . Dans ce cas, vous pouvez y apporter des modifications, même des modifications beaucoup plus sérieuses qu'une simple conversion en Norm.

▸ **Des pages** neuf

Cloner ce wiki localement

```
https://github.com/VBrazhnik/Corewar.wiki.git
```