

Explication du programme

Mohamed Hbaieb

15 mars 2025

1 Introduction

Ce programme ouvre un fichier, lit son contenu, et gère les erreurs potentielles telles que l'absence d'arguments, l'échec de l'ouverture du fichier, ou un fichier trop volumineux.

2 Structure du programme

Le programme est divisé en deux sections principales :

- **.text** : Contient le code exécutable.
- **.data** : Contient les données initialisées, comme les chaînes de caractères et les tampons.
- **.eqv maxchar 1024** : Définit une constante **maxchar** représentant la taille maximale du tampon d'entrée (**input**).

3 Fonctionnement du programme

Le programme commence à l'étiquette **main**. Voici les étapes principales :

3.1 Sauvegarde des arguments

- **mv s0, a0** : Sauvegarde le nombre d'arguments dans **s0**.
- **mv s1, a1** : Sauvegarde le tableau d'arguments dans **s1**.

3.2 Vérification des arguments

- **beqz a0, err_args** : Si aucun argument n'est fourni, le programme saute à **err_args**.
- **addi a1, a0, -1** : Soustrait 1 du nombre d'arguments.
- **bnez a1, err_args** : Si le nombre d'arguments n'est pas exactement 1, le programme saute à **err_args**.

3.3 Ouverture du fichier

- **lw a0, 0(s1)** : Charge le nom du fichier dans **a0**.
- **beqz a0, err_args** : Si le nom du fichier est vide, saute à **err_args**.
- **li a1, 0** : Mode d'ouverture (lecture seule).
- **li a7, 1024** : Charge le numéro de l'appel système **open**.
- **ecall** : Ouvre le fichier.
- **addi a2, a0, 1** : Vérifie si le descripteur de fichier est valide.
- **beqz a2, err_ouverture** : Si le fichier n'a pas pu être ouvert, saute à **err_ouverture**.
- **mv s0, a0** : Sauvegarde le descripteur de fichier dans **s0**.

3.4 Lecture du fichier

- **la a1, input** : Charge l'adresse du tampon d'entrée dans **a1**.
- **li a2, maxchar** : Charge la taille maximale du tampon dans **a2**.
- **li a7, 63** : Charge le numéro de l'appel système **read**.
- **ecall** : Lit le fichier.
- **sub a2, a2, a0** : Calcule le nombre de caractères lus.
- **mv s1, a0** : Sauvegarde le nombre de caractères lus dans **s1**.
- **mv a0, s0** : Restaure le descripteur de fichier dans **a0**.
- **li a7, 57** : Charge le numéro de l'appel système **close**.
- **ecall** : Ferme le fichier.

3.5 Vérification de la taille du fichier et fin

- `beqz a2, err_flong` : Si le fichier est trop gros pour le tampon, saute à `err_flong`.
- `b fin` : Saute à l'étiquette `fin` pour terminer le programme.

4 Gestion des erreurs

Le programme gère trois types d'erreurs :

- `err_args` : Affiche "erreur dans les arguments".
- `err_ouverture` : Affiche "erreur d'ouverture".
- `err_flong` : Affiche "fichier trop gros pour le tampon d'entrée".

5 Section .data

- `str_err_args` : Chaîne pour l'erreur d'arguments.
- `str_err_flong` : Chaîne pour l'erreur de fichier trop gros.
- `str_err_ouverture` : Chaîne pour l'erreur d'ouverture.
- `input` : Tampon de taille `maxchar` pour stocker le contenu du fichier.

6 Code source complet

Voici le code source du programme en assembleur RISC-V :

```
1 .text
2 .eqv    maxchar 1024
3
4 main:
5     # On sauve a0 et a1 dans s0 et s1 pour ne pas les perdre
6     mv s0, a0          # nombre d'arguments
7     mv s1, a1          # tableau d'arguments
8
9     beqz    a0, err_args
10    addi    a1, a0, -1
11    bnez    a1, err_args
12
13    lw a0, 0(s1)    # nom du fichier
14    beqz    a0, err_args
15
16    li a1, 0
17    li a7, 1024
18    ecall          # open(argv[1])
19    addi    a2, a0, 1
20    beqz    a2, err_ouverture    # erreur ouverture
21    mv s0, a0
22
23    la a1, input
24    li a2, maxchar
25    li a7, 63
26    ecall
27    sub a2, a2, a0
28    mv s1, a0
29    mv a0, s0
30    li a7, 57
31    ecall          # fermer
32
33    beqz    a2, err_flong    # assume que le fichier est trop long
34
35    ###
36    # charge utile - payload
37    ###
38    b fin
39
40 err_args:
41    la a0, str_err_args
42    b ecrire
43 err_ouverture:
44    la a0, str_err_ouverture
45    b ecrire
46 err_flong:
47    la a0, str_err_flong
```

```

48 ecrire:
49     li a7, 4
50     ecall
51 fin:
52     li a7, 10
53     ecall
54
55 .data
56
57 str_err_args: .string  "erreur dans les arguments"
58 str_err_flong: .string "fichier trop gros pour le tampon d'entree"
59 str_err_ouverture: .string "erreur d'ouverture"
60
61 input: .space maxchar

```