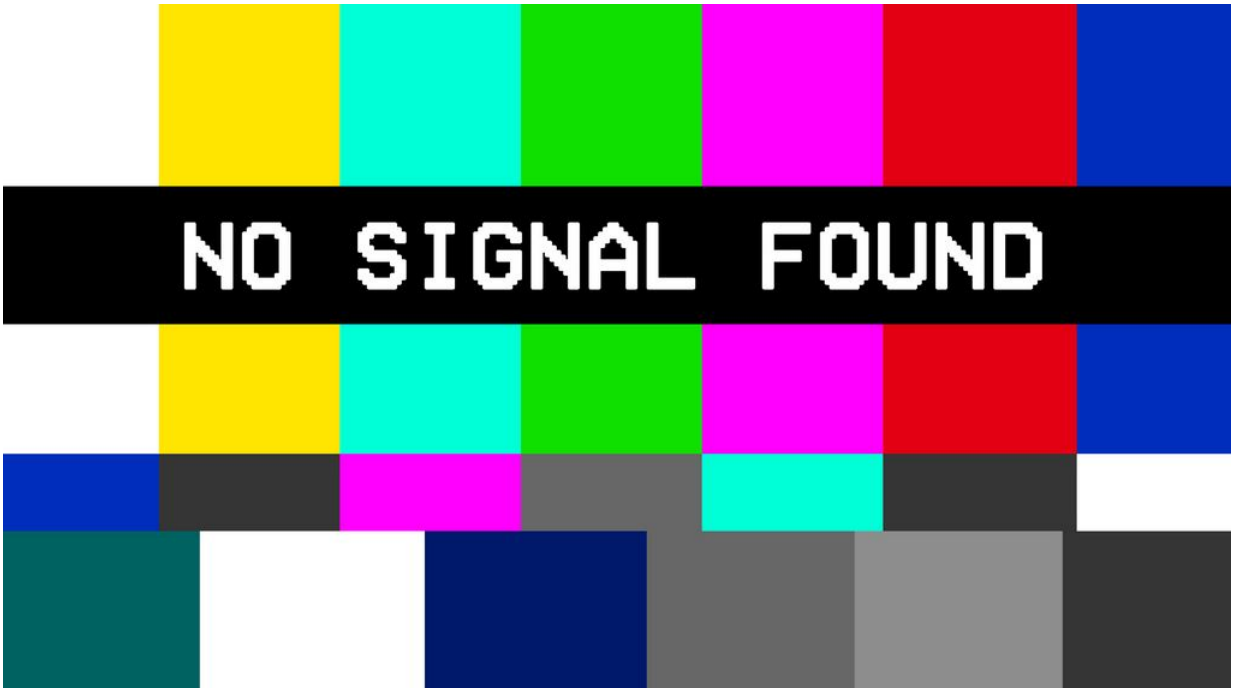
Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Systèm embarqués

Auteurs :
Marc Roten
Sven Rouvinez

Professeur :
Daniel Gachet



14 janvier 2019

Table des matières

1 Heure de travail

12 heures

2 Introduction

Dans ce travail nous allons réaliser notre premier TP en assembleur en réalisant une tour de Hanoi.

3 Synthèse

Sven ***Acquis***

- Appel de fonction en assembleur
- Passage de paramètres
- Sauvegarde des registres et restauration

Marc ***Acquis***

- Programmation en assembleur
- Debuggage assembleur
- Utilisation du Display du beaglebone
- Sauvegarde de registre et appel de fonction

4 Question

Pour les deux structures struct S1 et struct S2 et le code ci-dessous ? indiquez la convention utilisée pour :

```
struct S1 {int a;};
struct S2 {int a; int b[100];};

struct S1 f1();
struct S2 f2();

void f3 (int a, int b, int c, int d, struct S2 s);
void f4 (int a, int b, int c, int d, const struct S2* s);
void f5 (struct S2 s, int a, int b, int c, int d);
void f6 (const struct S2* s, int a, int b, int c, int d);
void f7 (struct S1 s);

void main(){
//point 1
struct S1 r1 = f1();
struct S2 r2 = f2();

//point 2
f3 (12,43,16,19,S2);
f4 (76,33,11,22,S2);
f5 (S2,11,22,33,44);
f6 (&S2,99,88,77,66);
f7 (&S1);
}
```

1. Le retour de ces structures par les fonctions f1 et f2
   - `f1()` Retourne uniquement un int contenu dans la structure S1, donc un seul registtre suffit et sera retournée par lui même
   - `f2()` Retourne un int et un tableau contenant 100 int. Un registre est limité à 32 bit donc il faudra retourne une adresse qui pointe sur la stack qui elle contient les 101 int
2. Le passage par valeur de ces structures par les fonctions f3 à f7 : les 4 premiers arguments sont donnés par les registres r0 à r3, et s'il y a plus de 4 arguments ils doivent être passés par la pile
   - `f3()` les 4 premiers int seront passés par les registres r0 à r3 et la struct sera passé par la pile
   - `f4()` les 4 premiers int seront passés par les registres r0 à r3 et le dernier argument est un pointeur constant sur la struct S2 et l'adresse est donnée par la pile

- `f5()` le passage des paramètres est mal agencé ici car la structure prend de la place et ne peut donc pas être placée dans le registre 0, par contre pour les autres paramètres il sera possible de les passer par les registres et un par la pile
- `f6()` étant donné que la structure est un pointeur et une constante, le registre 0 contiendra uniquement l'adresse, les 3 autres par les registres r1 r2 et r3 et le dernier par la pile
- `f7()` le seul argument est la structure S1 et elle contient uniquemement un int donc elle peut être passée par le r0 mais cela n'est pas optimal, par contre il ne sera pas possible de modifier la structure dans d'autres méthodes vu qu'elle est passé par copie et donc cela limite les effets de bords

5 Conclusion

Ce TP était intéressant, il nous a permis de comprendre comment traduire une série de fonction en assembleur et de pouvoir comprendre comment passer des paramètres ainsi que l'écriture de fonctions récursive. On a aussi pu constater que la taille (en byte) d'un fichier assembleur était bien moindre que la taille d'un fichier C. Ce qui est non-négligable pour des systèmes embarqués