



DIALLO AISSATOU BOBO

Groupe IE3-00-02

AHRICHE NAJAT

Groupe IE3-00-01

Compte rendu du Tp3
Elec3A

Exercice 1 :

Le code fourni copie une chaîne de caractères en assembleur ARM en utilisant un sous-programme. Le programme principal charge les adresses source et destination puis appelle le sous-programme avec l'instruction `BL recopies`.

Le sous-programme lit chaque octet de la chaîne source (`LDRB`), l'écrit dans la chaîne destination (`STRB`), puis avance les deux pointeurs. La copie s'arrête lorsqu'un octet nul 0 (fin de chaîne) est lu.

Enfin, l'instruction `MOV pc, lr` permet de revenir proprement au programme principal, ce qui valide le fonctionnement du sous-programme.

```
        AREA  Copy, CODE, READONLY
        ENTRY
        EXPORT __main

; PROGRAMME PRINCIPAL

__main
    ADRL  r1, srcstr    ; r1 → chaîne source
    ADRL  r0, dststr    ; r0 → chaîne destination

    BL    recopies      ; appel du sous-programme de copie

stop
    WFI                  ; fin du programme, attente d'interruption

; SOUS-PROGRAMME : recopies
; Copie la chaîne pointée par r1 vers celle pointée par r0
; Retour au programme principal avec MOV pc, lr

recopies
    LDRB  r2, [r1], #1   ; lit 1 octet depuis src → r2, avance r1
    STRB  r2, [r0], #1   ; écrit r2 dans dst, avance r0
    CMP   r2, #0         ; fin de chaîne atteinte ?
    BNE   recopies       ; si pas 0 → continuer
    MOV   pc, lr         ; retour au programme principal

; SECTION DONNÉES

        AREA  Strings, DATA, READWRITE

srcstr  DCB "premiere chaine - source", 0
dststr  DCB "deuxieme chaine - destination", 0

        END
```

Exercice 2 :

La fonction `strcpy` est écrite en assembleur pour être appelée depuis un programme en C. Le langage C envoie les deux paramètres dans les registres `r0` (destination) et `r1` (source), selon la convention ARM. Le sous-programme copie ensuite la chaîne source vers la chaîne destination octet par octet à l'aide des instructions `LDRB` et `STRB`. La copie se poursuit jusqu'à la rencontre du caractère nul 0, qui marque la fin de la chaîne. L'instruction `BX lr` permet enfin de revenir au programme C une fois la copie terminée.

Code :

```
AREA  recopie, CODE, READONLY
EXPORT strcpy
; void strcpy(char *d, char *s)
; Paramètres :
; r0 = d (destination)
; r1 = s (source)

strcpy
copie_octet
    LDRB  r2, [r1], #1    ; lire un octet dans la source (r1++)
    STRB  r2, [r0], #1    ; écrire cet octet dans la destination (r0++)
    CMP   r2, #0          ; fin de chaîne ?
    BNE   copie_octet     ; si non, continuer
    BX    lr              ; retour en C

END
```

Programme C appelant la fonction assembleur

```
#include <stdio.h>

extern void strcpy(char *d, char *s); // déclaration de la fonction assembleur

int main()
{
    char *srcstr = " premiere chaine - source ";
    char *dststr = " deuxieme chaine - destination ";

    printf("Avant recopie :\n");
    printf(" src = %s\n dst = %s\n", srcstr, dststr);

    strcpy(dststr, srcstr); // appel de la fonction assembleur

    printf("Après recopie :\n");
    printf(" src = %s\n dst = %s\n", srcstr, dststr);

    return 0;
}
```