

ÜBUNG: Parameterübergabe: „Call-by-value“ über Register (CbV-Reg)

Schreiben Sie ein Unterprogramm, welches die *Fakultät* $n!$ zu einer gegebenen Zahl n berechnet.

Verwenden Sie das Register $r0$ zur Übergabe des Eingabeparameters (n) und des Ergebnisses ($n!$).

Was sind die Beschränkungen dieser Methode?

Fakultät (In: n , Out: n Fakultät)

Zähler $\leftarrow 2$ // nur rechnen, wenn Input ≥ 2

Ergebnis $\leftarrow 1$

while (Zähler $\leq n$) do

Ergebnis \leftarrow Ergebnis \times Zähler

Zähler \leftarrow Zähler $+ 1$

end while

n Fakultät = Ergebnis

return

Ergebnis: $r1$

Zähler: $r2$

AsmTest_010

```
        AREA MyCode, CODE, readonly, align = 3

;-----
; main subroutine
;-----
        EXPORT main [CODE]

main     PROC

        BL      Init_TI_Board ; Initialize the serial line to TTY
                                ; for compatability to out TI-C-Board

        mov     r0, #7        ; Input-Parameter
        bl      Fakultaet     ; Fakultaetsfunktion

forever  b       forever      ; Programmende (Endlosschleife)
; -----
; Ende : main
; -----

        ALIGN

; -----
; Fakultaet (Unterprogramm)
; r0: In (=n), Out(=n!)
; r1: Ergebnis
; r2: Zaehler
; -----
Fakultaet
        push    {r1-r3, lr}   ; Reg. retten

        mov     r1, #1        ; Ergebnis init.
        mov     r2, #2        ; Zaehler init.

while_01
        cmp     r2, r0        ; Abbruch, wenn Zaehler > n
        bhi     endwhile_01

do_01
        mul     r1, r2, r1    ; Erg. = Erg. * Zaehler
        add     r2, #1        ; Zaehler = Zaehler + 1

        b       while_01
endwhile_01

        mov     r0, r1        ; Ergebnis in Ausgaberegister kopieren

        pop     {r1-r3, lr}   ; Reg. restaurieren
        bx      lr           ; return zum aufrufenden Programm
; -----
; Ende : Fakultaet
; -----

        ENDP
        ALIGN

        END
```

ÜBUNG: Parameterübergabe: „Call-by-reference“ über Register (CbR-Reg)

Schreiben Sie ein Unterprogramm, welches die Länge eines Strings bestimmt.

Zur Übergabe des Strings soll das Register r0 verwendet werden.
Das Ergebnis soll ebenfalls in r0 stehen.

Was sind die Beschränkungen dieser Methode?

StrLen (In : Adresse des Strings , Out : Stringlänge)

Stringlänge $\leftarrow 0$

Zeichenpositionszeiger (ZPZ) \leftarrow Stringadresse

Aktuelles Zeichen \leftarrow Wert, auf den der ZPZ zeigt

while (Aktuelles Zeichen $\neq 0$) **do**

Stringlänge \leftarrow Stringlänge + 1

ZPZ = ZPZ + 1

Aktuelles Zeichen \leftarrow Wert auf den der ZPZ zeigt

endwhile

return

ZPZ : r0

Stringläng : r1

Aktuelles Zeichen : r2

AsmTest-011

```
        AREA MyData, DATA, align = 2
        GLOBAL MyData, MyText

MyText      DCB  "123456789",0

;*****
; Code section, aligned on 8-byte boundary
;*****

        AREA MyCode, CODE, readonly, align = 3

;-----
; main subroutine
;-----
        EXPORT main [CODE]

main      PROC
        ldr      r0, =MyText          ; Stringadresse übergeben
        bl       StrLen              ; Stringlaenge bestimmen

; Programmende (Endlosschleife)
loop      b       loop

; -----
; Ende : main
; -----

        ALIGN

; -----
; Stringlaenge bestimmen (Unterprogramm)
; r0: In (=n), Out(=n!)
; r1: Ergebnis
; r2: Zähler
; -----
StrLen    push    {r1-r3, lr}          ; Register retten

        mov      r1, #0              ; Stringlaenge init.

while_01  ldrb     r2, [r0], #1         ; Aktuelles Zeichen lesen, ZPZ = ZPZ+1
        cmp      r2, #0              ; Abbruch, wenn akt. Zeichen = 0 (Stringende)
        beq      endwhile_01

do_01     add      r1, #1              ; Stringlaenge incr.
        b       while_01

endwhile_01

        mov      r0, r1              ; Ergebnis in Ausgaberegister kopieren

        pop      {r1-r3, lr}         ; Register restaurieren
        bx       lr                  ; return zum aufrufenden Programm

; -----
; Ende : StrLen
; -----

        ENDP
        ALIGN

        END
```