

# AAPCS

- *ARM Architecture Procedure Call Standard*
- Passagem de parâmetros para uma função:
  - Primeiros parâmetros em R0, R1, R2 e R3;
  - Demais parâmetros pela pilha.
- Retorno da função:
  - R0 (32 bits)
  - R1:R0 (64 bits)
- Alinhamento da pilha deve ser 64 bits:
  - PUSH/POP de número par de registradores antes/depois de chamadas de funções

# Chamando uma função em C do ASM

```
THUMB
    IMPORT Conta
    IMPORT A
    AREA |.text|,CODE,READONLY,ALIGN=2
    EXPORT Start

Start
    LDR R2, =A      ;R2 = &A
    MOV R0, #2
    STR R0, [R2]   ;A=2

Loop
    MOV R0, #5     ;Função recebe
                  ;parâmetros em R0 a R3
    BL Conta      ;Chama a função que irá
                  ;multiplicar A (variável
                  ;global) pelo parâmetro 1
                  ;que está em R0 e coloca o
                  ;resultado em R0

    NOP

    ALIGN
    END
```

```
#include <stdint.h>

uint32_t A;

//Multiplica o parâmetro pelo fator A
uint32_t Conta(uint32_t parametro)
{
    uint32_t resultado;
    resultado = A * parametro;
    return resultado;
}
```

# Chamando uma função em ASM do C

```
THUMB
    AREA DATA, ALIGN=2
    EXPORT A [DATA, SIZE=4]
A    SPACE 4
    AREA |.text|, CODE, READONLY, ALIGN=2
        EXPORT Conta
        EXPORT A

;Retorna a multiplicação de A que está na
;memória pelo parâmetro recebido em R0 e
;retorna em R0 para a AAPCS
Conta
    LDR R2, =A      ;R2 = &A
    LDR R1, [R2]    ;R1 = [A]
        MUL R0, R0, R1
    BX LR

    ALIGN
    END
```

```
#include <stdint.h>

uint32_t Conta(uint32_t parametro);
extern uint32_t A;

int main (void)
{
    uint32_t resultado;
    A = 2;

    while (1)
    {
        resultado = Conta(5);
        A = resultado;
    }
}
```

# Exemplo

- Abrir o projeto exemplo em C (anexo)
- *Build* e executar passo-a-passo