

14M. 14553120. 焦子鸣. WSD5

Q1: 用 R0, R1, R2 传递数字.

Q2: 用 R0, R1, R2, R3 传递数字,
最后一个数先放在栈中.

Q3: 应该将其存在栈中, 并在最后
恢复其初始值.

Q4: 使用 R12 的时候不应该保存.

Q5: 由题意可得:

AA SPACE 10 ; 分配 10 个字节存储空间.

BB RN 2 ; 指定 BB 标签为 R2 寄存器

CC DCB 1, 2, 3 ; 定义 3 个字节存 1, 2, 3.

DD DCB "JohnInr", 0 ; 定义字符串存 null

EE DCW 1, 2, 3 ; 定义 3 个半字存 1, 2, 3

FF PCD 1, 2, 3 ; 定义 3 个半字存 1, 2, 3

GG EQU 10 ; 定义 GG 作为 10 的符号.

EXPORT Fun ; 使 "Fun" 可被其它文件使用

IMPORT Happiness ; 链接器-文件中的 "Happiness"

AREA DATA, ALIGN=2 ; 放入 RAM 中

AREA ltext1, CODE, READONLY, ALIGN=2 ;
放入 ROM 中.

Q6: C: ~~uint8_t~~ const Table[256] = {0, 1, 1, ..., 2,
2, ..., 3, 3, ..., 4, ..., 5, ..., 6, ..., 7, ..., 8, ..., 9, ...,
10, ..., 11, ..., 12, ..., 13, ..., 14, ..., 15, ..., 16, ...};
~~uint8_t~~ Sqrt(~~uint8_t~~ n) {
return Table[n]
}

汇编: Table DCB 0, 1, ..., 2, ..., 3, ..., 4, ...,
DCB 5, ..., 6, ..., 7, ..., 8, ..., 9, ..., 10, ...,
DCB 11, ..., 12, ..., 13, ..., 14, ..., 15,
DCB 16, ...

Sqrt LDR RL, Table
LDRB R0, [R1, R0]
BX LR

~~uint8_t~~ const Table[256].

Q7:

汇编:

Count MOV R1, R0

MOV R0, #0

Loop LDRB R2, [R1]

CMP R2, #0

BEQ Done

ADD R1, #1

ADD R0, #1

B Loop

Done EX LR.

C:

uint32_t Count(char* p) {

uint32_t n=0;

while (*p) {

p++;

n++;

return n;

}

Q8: 汇编:

AREA DATA, ALIGN=2

Bnf SPACE 40

AREA ltext1, CODE, READONLY, ALIGN=2

set LDR R1, Bnf

MOV R0, #0

STR R0, [R1]

ADD R1, #1

ADD R0, #1

CMP R0, #10

BLO Loop

EX LR.

Q8: C:

```
uint32_t Buf[10];  
void set(void) {  
    uint32_t n;  
    for (n=0; n<10; n++) {  
        Buf[n] = n;  
    }  
}
```

Q9:

传值的~~等~~情况为:

- I. ^是只传数据到函数中
- II. 传入的数据量很少
- III. 不想传入的数据被修改

传引用情况为:

- I. 要传入的数据非常大
- II. 希望函数修改原数据内容
- III. 希望保存一个数据的共享副本