



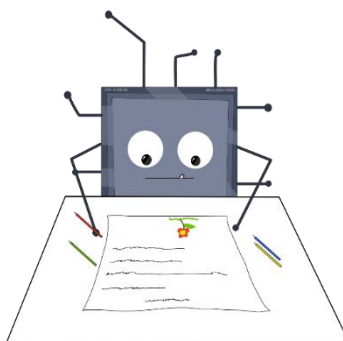
Homework 2

Lec 13-25

- ◀ مهلت ارسال تمرین ساعت ۲۳:۵۹ روز ۱۹ دی می باشد.
- ◀ سوالات خود را می توانید از طریق تلگرام از آقای سپهر مقیسه (@Ihavegoosebumps)، آقای فراز فرنگی - زاده (@faraz_ffff)، خانم زهره کریمی (@zohreh_karimi01)، خانم محدثه اطمیابی (@Mohadeseh_atyabi)، خانم روژینا ستارپور (@Rozhiiiiin_S) و خانم زهرا رحیمی (@Zahra_Rahimii) بپرسید.
- ◀ برای ارائه پاسخ پرسش های عملی بهتر است نکات زیر رعایت شود:
 - (۱) ارائه اسکرین شات از کد و نتیجه اجرای آن در یک فایل pdf.
 - (۲) قرار دادن فایل کد و اسکرین شات از نتیجه اجرای کد. در صورت استفاده از این روش حتما هر سوال را در پوشه ای جداگانه قرار دهید.
- پرسش های عملی را با استفاده از Keil انجام دهید: سوال ۳، سوال ۴، سوال ۶
- ◀ فایل پاسخ تمرین را تنها با قالب HW2_StudentNumber.pdf و یا HW2_StudentNumber.Zip در مودل بارگزاری کنید.

- نمونه: HW2_9831090.pdf

- نمونه: HW2_9831090.Zip





سوال ۱:

فرض کنید مقداری که r3 به آن اشاره دارد به این صورت است:

0x45
0xA5
0xC6
0x71

ثبات r3

و این ثبات در خانه 0x2000 0300 تا 0x2000 0303 حافظه قرار دارد.

ثبات r1 و r2 به ترتیب دارای شکل مشخص شده زیر هستند. همچنین ثبات r4 و r5 و r6 به ترتیب آدرس r1 و r2 و r3 را در خود دارند.

0xF5	0x34	0x21	0x56
------	------	------	------

ثبات r1

0x87	0xA4	0x5C	0x38
------	------	------	------

ثبات r2

به سوالات زیر پاسخ دهید (مقادیر اولیه ثبات ها و استک در هر بخش همان مقدار اولیه سوال است):

الف) مقدار r3 در پایان اجرای قطعه کد زیر چیست؟

```
strb r2 [r6];  
ldrb r1 [r6];  
ldrh r2 [r4];  
ldr r3 [r5];  
strh r2 [r4];  
ldrb r3 [r4];
```



ب) توضیح دهید تفاوت `ldr-str` و `mov` چیست و به نظر شما کدام از نظر کارایی سریع تر است؟ چرا؟

ج) فرض کنید یک دستور فرضی جدید به صورت `ldrq` داریم که $\frac{3}{4}$ یک کلمه را می تواند جابه جا کند و کاملاً در جای دیگر بریزد (به این معنی که $\frac{3}{4}$ کلمه از خانه ای که برداشته می شود، خالی می گردد و در مکان جدید ریخته می شود). چگونه می توان با حداقل دستورات ممکن و به کمک `ldrq`، مقادیر `r1` را با `r2` جابه جا کرد؟ (استفاده از سایر دستورات مانند `mov` نیز جایز بوده به شرط آن که مسقیماً برای خواسته اصلی سوال استفاده نشود).

به طور مثال یک ثابت داریم با مقدار:

`r=0x43F598C1`

بعد از اجرای دستور `ldrq r2 [address of r]` مقدار جدید به این صورت می باشد:

`r2 = 0x87F598C1` و `r=0x43`

و حال اگر بر روی `r`، `mov` را اجرا کنیم مقدار `0x43` در خانه اول ثابت مقصد قرار میگیرد و سایر خانه ها دست نخورده باقی می ماند.

راهنمایی: از `r3` برای جابه جایی کمک بگیرید.

توجه: در ثابت های `r4` و `r5` چیزی نباید بریزید چرا که ادرس ثابت های `r2` و `r1` در آنها هستند.

د) با فرض وجود دستور `ldrq` با تعریف سوال قبل خروجی قطعه کد زیر را مشخص کنید:

```
str  r1 [r6];  
ldrb r2 [r4];  
strh r3 [r5];  
ldrq r1 [r5];  
strb r2 [r6];  
ldrq r3 [r4];
```



سوال ۲:

با توجه به کد روبرو به سوالات زیر پاسخ دهید.

```
Area Code_Section, Readonly, Code
    LDR R3, =Q_Data;
    MOV R4, #'2';
    SUB R4, R4, #0x29;
    ADD R3, R3, R4;
    LDRB R5, [R3];
    HERE B HERE

Area Data_Section, Data
Q_Data
    DCD 0x18
    DCB "HWDB"
    ALIGN 2
    DCB 0x36, 0x10
    END
```

الف) پس از اجرای برنامه، محتوای نهایی رجیستر R5 را در حالتی که اعداد با متد Little Endian در رجیسترها ذخیره شوند به دست آورید.

ب) در صورتی که دستور ALIGN 2 را کامنت کنیم، نگاشت حافظه چه تغییری خواهد کرد؟ محتوای نهایی رجیستر R5 در این صورت چه خواهد بود؟

ج) آیا می توان شبه دستور LDR در خط دوم را با دستور دیگری جایگزین کرد؟



سوال ۳:

فرض کنید مقدار اولیه رجیستر R10 معادل باینری شماره دانشجویی شما، R0 و R1 دو عدد دلخواه، R4 آدرس حافظه اولین گره در یک لیست پیوندی از اعداد صحیح و R5 آدرس اول لیست پیوندی می باشد. برنامه ای بنویسید که مشخص کند چه تعداد الگوی "۱۰۱" در رجیستر R10 وجود دارد. (مثال: باینری ۹۸۳۱۰۹۰ مقدار ۱۰۱۰۱۱۰۰۰۰۰۰۱۰۱۰۱۱۰۱۰ است که الگوی ۱۰۱ در آن ۳ بار تکرار شده است).



سوال ۴:

برنامه‌ای بنویسید که حاصل جمع دو آرایه از integer ها را با استفاده از subroutine پیدا کند. توجه داشته باشید که آرایه‌های ورودی signed و ۱۶ بیتی هستند. جواب آرایه اول را در POUT1&POUT2 و جواب آرایه دوم را در POUT3&POUT4 نشان دهید.



سوال ۵:

۴ مورد از قوانین استاندارد AAPCS را برای پیاده‌سازی توابع شرح دهید.



سوال ۶ (امتیازی):

به کمک زبان اسمبلی یک ماکرو به نام FuncW بنویسید که معادله زیر را حل کند:

$$W = 2X + 5Y - 4Z - 6$$

ماکرو باید ۴ پارامتر را قبول کند. اولین پارامتر varW است که بیانگر W بوده که همان رجیستر جواب است. پارامترهای دوم تا چهارم نیز varX، varY، varZ بوده که بیانگر ورودی‌های X، Y، Z هستند. این سه متغیر می‌توانند هر رجیستری باشند. همچنین لیبل ماکرو باید solveW باشد. ماکرو بیان‌شده را پیاده‌سازی کنید.