

به نام خدا



درس ساختار و زبان کامپیوتر
نیم سال اول ۰۴-۰۳
استاد: دکتر اسدی

دانشکده مهندسی کامپیوتر

تمرین سری سوم

- پرسش‌های خود را در سامانه CW و تالار مربوط به تمرین مطرح نمایید.
- پاسخ سوالات را تایپ نمایید.
- اسکرین‌شات‌ها، عکس‌ها، فایل‌های مربوط به سوال عملی، گزارش تمرینات عملی و PDF قسمت تئوری را در پوشه با نام به فرمت HWNUM_StudentID1_StudentID2 ذخیره نمایید. سپس آن را zip نمایید و در صفحه درس بارگذاری نمایید. به عنوان مثال یک فایل بارگذاری شده قابل قبول باید دارای فرمت HW1_400123456_403123456.zip باشد.
- هر دانشجو می‌تواند حداکثر دو تمرین را با دو روز تأخیر بدون کاهش نمره ارسال نماید.
- تمرینات عملی به صورت گروه‌های دو نفر تحویل داده شود.
- هر دو عضو گروه موظف هستند تمرینات خود را بارگذاری کنند.
- عواقب عدم تطابق بین پاسخ دو عضو گروه برعهده خودشان است.
- تحویل تمرین به صورت انگلیسی مجاز نیست. در صورت تحویل تمرین به صورت انگلیسی (حتی بخشی از تمرین) نمره تمرین موردنظر صفر در نظر گرفته می‌شود.
- در صورت مشاهده تقلب برای بار اول نمره هر دو طرف صفر می‌شود. در صورت تکرار نمره کل تمرینات صفر خواهد شد.
- استفاده از ابزارهایی مانند ChatGPT به منظور ابزار کمک آموزشی مجاز است به شرط آن که به خروجی آن اکتفا نشود.
- توجه شود که پروژه نهایی درس در گروه‌های چهار نفر تحویل گرفته می‌شود.
- سوالات با عنوان اختیاری نمره‌ای ندارند اما جواب دادن به آن‌ها کمک به سزایی در یادگیری درس می‌کند.

تمارین تئوری

۱. به سوالات زیر پاسخ دهید:

- (آ) درباره یک شیوه در نوشتن کدهای اسمبلی به نام کد خودتغییردهنده^۱ تحقیق کنید و نتیجه تحقیق خود را با ذکر مثالی ساده شرح دهید.
- (ب) درباره فراخوان سیستمی^۲ تحقیق کنید و بگویید چرا برخی از کارکردهای سیستمی، به صورت فراخوان سیستمی مورد استفاده قرار می‌گیرند.
- (ج) مقادیر پارامترهای برخی از فراخوان‌های سیستمی در اسمبلی MIPS مانند خروج^۳، نوشتن^۴ و خواندن^۵ را بیابید.
- (د) کد زیر را بررسی کنید و بگویید که مقادیر نهایی ثابت‌های \$a0 و \$a1 چقدر خواهد بود. (مراحل را گام به گام توضیح دهید.)

```

1 .data
2
3 .text
4 .globl main
5
6 main:
7     li $a0, 5
8     li $a1, 8
9
10    jal func
11
12    jal exit
13
14 exit:
15    li $v0, 10
16    syscall
17
18 go_to_exit:
19    j exit
20
21 func2:
22    la $t6, go_to_exit
23    lw $t4, 0($t6)
24    la $t6, jump_section
25    sw $t4, 0($t6)
26 func:
27    add $a0, $a0, $a1
28    la $t9, func
29    lw $t8, 0($t9)
30    li $t7, 2048
31    xor $t8, $t7, $t8
32    sw $t8, 0($t9)
33 jump_section:
34    j func2

```

^۱Self-modifying code

^۲Syscall

^۳exit

^۴write

^۵read

(ه) با توجه به اطلاعاتی که درباره فراخوان سیستمی کسب کردید، دو خط ۷ و ۸ را به گونه‌ای تغییر دهید که مقادیر دو ثبات، از کاربر ورودی گرفته شوند.

۲. خروجی کد زیر چه خواهد بود و علت آن را توضیح دهید.
راهنمایی: تابع `print_int` عدد صحیحی که در ثبات `a0` می‌باشد را چاپ می‌کند. دستور `jal` برای فراخوانی توابع استفاده می‌شود.

```

1 .data
2 a: .word 2
3
4 .text
5 main:
6     la $s0, a
7     la $s1, op
8     sub $s0, $s1, $s0
9     sub $s1, $zero, $s0
10    lw $s0, a($s0)
11    lw $s1, op($s1)
12    add $s0, $s0, $s1
13    sw $s0, op($zero)
14    la $s0, label
15    la $s1, op
16    sub $s0, $s1, $s0
17    sw $s0, a($zero)
18 label:
19    li $s0, 10
20    li $s1, 15
21 op:
22    add $a0, $s0, $s1
23    jal print_int
24    lw $a0, a($zero)
25    jal print_int
26    jal end
27
28 print_int:
29    li $v0, 1
30    syscall
31    jr $ra
32
33 end:
34    li $v0, 10
35    syscall

```

۳. فرض کنید در شبکه‌ای مشابه، دو دستگاه A و B در حال ارسال و دریافت داده هستند. هر دستگاه از یک نوع اندین مختلف استفاده می‌کند. دستگاه A از "big endian" و دستگاه B از "little endian" استفاده می‌کند. داده‌های دستگاه A باید از چندین گره عبور کنند، به گونه‌ای که هر گره می‌تواند اندین خود را داشته باشد و این اندین‌ها به طور تصادفی تغییر می‌کنند.

(آ) در یک شبکه دستگاه A مقدار ۶۴ بیتی 0x123456789ABCDEF0 را ارسال می‌کند. این مقدار باید از ۴ گره عبور کند که هر کدام اندین متفاوتی دارند. ترتیب اندین‌ها در هر گره به این شکل است: گره اول: little endian - گره دوم: big endian - گره سوم: little endian - گره چهارم: big endian، مقدار نهایی‌ای که به دستگاه

B با اندین "little endian" ارسال می‌شود چیست؟ مراحل محاسباتی تبدیل در هر گره را به طور دقیق شرح دهید و مشخص کنید چگونه مقدار اولیه تغییر می‌کند.

(ب) فرض کنید در گره سوم، تبدیل اندین به درستی انجام نمی‌شود و به اشتباه مقدار به عنوان "big endian" پردازش می‌شود. توضیح دهید چه مقدار نهایی به دستگاه B خواهد رسید و چگونه این خطا باعث مشکلات در برنامه‌نویسی سیستم B می‌شود.

(ج) با تحقیق در مورد پروتکل‌های شبکه‌ای مثل TCP/IP، توضیح دهید که چگونه این پروتکل‌ها با مشکل تفاوت اندین در سیستم‌های مختلف مقابله می‌کنند. آیا در سطوح مختلف شبکه (مثل لایه انتقال یا لایه کاربرد) این مشکل به شکل متفاوتی حل می‌شود؟ مثال بزنید که چگونه یک نرم‌افزار می‌تواند به درستی داده‌ها را از سیستمی با معماری متفاوت دریافت کند.

یک سیستم MIPS که دارای آدرس‌دهی بایتی است و در حالت *BigEndian* و *LittleEndian* عمل می‌کند را در نظر بگیرید. یک عدد صحیح ۳۲ بیتی $0xA1B2C3D4$ داده شده که باید از آدرس $0x1000$ ذخیره شود.

۴. (آ) توالی بایت‌ها را توضیح دهید که چگونه در حافظه در آدرس‌های $0x1000$ ، $0x1001$ ، $0x1002$ و $0x1003$ در حالت *BigEndian* ذخیره می‌شود.

(ب) حالا، اگر عدد صحیح ۳۲ بیتی در آدرس $0x1000$ ذخیره شود، اما با یک عدد صحیح ۱۶ بیتی $0x1234$ که درست بعد از آن در آدرس $0x1004$ قرار دارد، توضیح دهید که چگونه کل توالی بایت‌ها در حافظه در آدرس‌های $0x1000$ تا $0x1005$ خواهد بود.

(ج) اگر سیستم در حالت *LittleEndian* باشد، توصیف کنید که چگونه توالی بایت‌ها در آدرس‌های $0x1000$ ، $0x1001$ ، $0x1002$ و $0x1003$ متفاوت خواهد بود.

(د) بحث کنید که چگونه وجود عدد صحیح ۱۶ بیتی $0x1234$ ذخیره شده در آدرس $0x1004$ بر روی آرایش در حالت *LittleEndian* تأثیر می‌گذارد، به ویژه در آدرس‌های $0x1004$ و $0x1005$.

(ه) پس از ذخیره عدد صحیح در حالت *BigEndian*، اگر سیستم دیگری که به صورت *LittleEndian* پیکربندی شده است، مقدار را از مکان حافظه $0x1000$ بخواند، توضیح دهید که چه مقدار نادرستی ممکن است تفسیر شود. توضیح دهید که توالی بایت‌هایی که خواهد دید چیست و چرا منجر به سوء تفاهم مقدار ذخیره شده می‌شود.

۵. کد زیر را بررسی کنید و توضیح دهید که بیت علامت ثبات $t5$ در نهایت چه چیزی را نشان می‌دهد؟ این بیت چگونه با بخش‌پذیری مقدار $number3$ بر ۴ ارتباط دارد؟ فرض کنید $number3$ یک عدد دلخواه ذخیره‌شده در حافظه است.

```

1 .data
2 number1:
3     .word 3
4 number2:
5     .word 1
6 number3:
7     .word 54    # arbitrary value, can be any number
8
9 .text
10 .globl main
11 main:
12     la    $t0, number1
13     la    $t1, number2
14     la    $t2, number3
15
16     lw    $t3, 0($t0)
17     lw    $t4, 0($t1)
18     lw    $t5, 0($t2)
19
20     and   $t5, $t5, $t3

```

```

21      sub $t5, $t5, $t4
22
23      jr  $ra          # Return from the main function

```

۶. ماشین هایی داریم که وضعیت Endianness شان نامشخص است. برای تشخیص این موضوع عملیات های ریاضی زیر را انجام داده ایم و نتایج مقابل را بدست آورده ایم. برای هر عملیات مشخص کنید وضعیت Endian به چه صورت است. اعداد نوشته شده از چپ به راست بایت های اول و دوم اعداد هستند.

71 42	+	32 c2	a4 04
e9 b6	+	3a cb	23 82
1d d5	-	88 41	95 93
ad ee	-	60 02	4d ec

تمارین عملی

نکات قابل توجه

- کد پاسخ سوالات عملی را با فرمت Q#.asm در پوشه‌ای به اسم practical قرار دهید.
- به ازای هر سوال دو فایل با اسامی input#.txt و output#.txt قرار دهید که ورودی نمونه سازگار با کد شما و خروجی انتظاری آن است.
- خط ابتدایی هر فایل کد، ورودی و خروجی شماره دانشجویی هر دو فرد را با یک فاصله کامنت کنید.
- فایل کد شما باید توسط کامند زیر نیز قابل اجرا باشد:

```
1 java -jar Mars.jar code.asm
```

۱. برنامه‌ای بنویسید که عملیات‌های جمع و تفریق را برای اعداد صحیح انجام دهد. دقت کنید که ورودی‌ها و خروجی‌های برنامه، فقط اعداد ۳۲ بیتی می‌باشند و overflow نداریم. (در این سوال برای عملیات‌های ریاضی، دستورات add و sub، برای عملیات‌های بیتی، دستورات and، or، nor و xor مجاز می‌باشند)
عدد اول در خط اول، نوع عملیات در خط دوم و عدد دوم در خط سوم به برنامه ورودی داده می‌شود. نمونه‌ای از ورودی و خروجی را در زیر می‌توانید مشاهده کنید.

```
1 input:
2 25
3 +
4 36
5 output:
6 61
```

```
1 input:
2 -17
3 -
4 5
5 output:
6 -22
```

۲. برنامه‌ای به زبان MIPS Assembly بنویسید که ضرایب یک معادله درجه دوم را در ورودی بگیرد و جواب‌های حقیقی این معادله را بصورت اعداد اعشاری در خروجی چاپ کند. فرض کنید حتماً معادله دو جواب حقیقی خواهد داشت. معادله به این صورت خواهد بود:

$$a \times x^2 + b \times x + c = 0$$

در سه خط متفاوت اعداد a، b، و c به ترتیب داده می‌شوند. سپس در خطوط جدا باید دو جواب حقیقی برای این معادله چاپ شود.

می‌توانید از این معادله برای پیدا کردن پاسخ‌ها استفاده کنید:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

۳. با استفاده از دستورات lw و sw و دستورات منطقی‌ای که تا کنون یاد گرفته‌اید، یک برنامه اسمبلی MIPS بنویسید که یک آرایه به طول ۷ را از حافظه بخواند و در صورتی که آرایه متقارن^۷ باشد، ثابت a0 را یک کند و در غیر این صورت ثابت a0 را صفر کند. فرض کنید بخش^۸ داده برنامه به صورت زیر است:

palindrome^۷
segment^۸

```
1 .data
2     array: .word 10, 20, 30, 40, 50, 60, 70    # arbitrary array of length 7
3     length: .word 7
4     one: .word 1
```

۴. بدون استفاده از هیچگونه دستور پرش، کد اسمبلی‌ای بنویسید که بیشینه مقادیر دو خانه `num1` و `num2` در حافظه را پیدا کند و مقدار آن را در خانه `max_value` در حافظه بریزد. برای نوشتن کد این سوال تنها مجاز به استفاده از دستورات `addi` و `xor` هستید. همچنین توجه داشته باشید که مقادیر ابتدایی داخل ثبات‌ها * نیست.