

امتحان نهایی سخت‌تر و زیاده‌تر ۱۲۶-۱۳۰

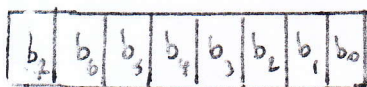
نیت: دوستانت

۱- توضیح دهید چرا روند افزایش فرکانس کاری پردازنده کند و متوقف و تبدیل به روند تثبیت هسته‌ای پردازشی گراید.

۲- دلیل اضافه کردن دستورالعمل جدید چیست؟ (MMX) و یا آرایه (SSE) به مجموعه دستورالعمل‌های سخت‌افزاری چه منافع و مزایایی به همراه داشت؟

۳- مثالی از دسترسی/مستطانی یک برنامه به بلوک (یا صفحات) حافظه نشان (یا مجازی) بزنید که در آن سیاست دیگری (مثل MRU, FIFO, ...) بر LRU (Least Recently Used) برتری داشته باشد (یعنی تعداد دفعات جایگزینی یک بلوک (صفحه) به خاطر ورود یک بلوک (صفحه) جدید در حافظه نشان (یا مجازی) کمتر از LRU باشد).

۴- ۱- قالب نمایش منبر ثابت مقابل داده شده است:



منبر

ن: بیت در موقعیت نام

استخوان‌کننده فرکانس عدد مثبت و کوچکترین عدد منفی (با زنی اینکه b_7 بیت علامت است) در نمایش شکل ۲ بیت.

۴- ۲- قدرت تفکیک این نمایش چیست (= Resolution = کوچکترین فاصله دو عدد استخوانی)
۴- ۳- دقت نمایش این کیبده را با حالتی که منبر بین b_6 و b_7 قرار بگیرد و در حالت کلی تر منبر کمتر مقایسه و نقد کنید.

۵- ۱- دستور ADC (Add with carry) یک جمع (مثلاً ۱۶ بیتی) بین دو عملوند A و B (مثلاً AX و BX) انجام می‌دهد یعنی اینکه بیت نفکی را نیز در این جمع وارد می‌کند. به کمک این دستور یک برنامه ساده بنویسید که یک جمع ۳۲ بیتی انجام دهد.

۵- ۲- دستور SBB (Subtract with borrow) را برای تفریق EAX-EBX بنویسید. به صورت یک سری تفریق ساده تر ۱۶ بیتی برنامه نویسی کنید.

۴-۱. فرض کنید می خواهیم مدخل دستور انتخاب شرطی (Jump if greater) (JG) بنویسیم (که مقدراً بعد از مقایسه دو عدد علامت دارند نشانی کند) به کمک دستور

با به شکل JZ، JC، JS و JO (Jump if overflow) (Jump if sign=1) بنویسید

کنیم. برنامه را به این منظور به طرز زیره بنویسید.

۴-۲. اگر خواستیم به جای JO مدخل پیدا کنیم چه دستوراتی را باید اجرا کردیم.

۷-۱. مشخص کنید اگر $\{CS=028E, IP=0040\}$ آدرس خطی واقعی ۲۰ بیت به Her چیست؟

۷-۲. حال اگر آدرس واقعی (Hex) 02640 را داشته باشیم نشان دهیم چندین زوج از آدرس ۶/ (CS:IP) ممکن است این را به وجود آورده باشند.

۸-۱. یک شکل نمادین از روش کار تبدیل یک آدرس منطقی به خطی در سیستم در حالت قطعه بندی شده (Segmented) با به کارگیری یک بیت Segment selector، Descriptor table و آدرس ۳۲ بیت (offset)، با ذکر عناصر جدول Descriptor ارائه دهید.

۸-۲. مکانیزم حفاظت Protection به چه دردی می خورد و در حافظه چگونه بنویسد؟

آدرس	داده
0	Data 1 DW 2000H
2	Data 2 DW 3000H
...	...
17	Les si, Data1
1A	mov Di, offset Data2
1D	mov Bx, [SI]
1F	mov CX, [DI]
21	mov [SI], CX
23	mov [DI], Bx
	.Exit
	END

۹-۱. این برنامه چه کاری انجام می دهد و چگونه؟

۹-۲. به نظر شما چرا وقتی دستور Les است از سبب دستور (Directive) البت offset استفاده شده است؟

۹-۳. با توجه به آدرس هر دستور ارائه شده در حافظه مشخص کنید اگر بخواهیم یک نقطه توقف (Breakpoint) بگذاریم چه کار می کنیم (mov cx, [DI]) قرار دهیم چه کار و چگونه باید بنویسیم.

۹-۴. اجرا بر روی برنامه در مایکروسافت استپ چگونه می باشد؟

مستند شده

۲۸

۱- روند افزایش فرکانس پردازنده؟ به این دلیل گفته شد که توان مصرفی (و حرارت تلف شده) آن با توان دو آن افزایش پیدا می کرد و لذا هم به شدت داغ می گشت و هم برق زیادی مصرف می کرد. لذا برای افزایش سرعت پردازنده؟ ترسیم داده گاه گاه به پردازشی می توانی مقدار در ولتاژ و فرکانس کمتر استفاده کنی که توان مصرفی و حرارت کمتری داشته باشند.

۲- اهداف گسترش دستور (MultiMedia Extension) (MMX) و Streaming SIMD Extension (SSE)

برای به کارگیری موازات اجرا در داخل پردازنده برای کارهای گوناگون و متنوع و درایه بود به گونه ای که هزینه های کمتری را در بر داشته باشد. این کار با استفاده از واحدهای مختلف و درایه بود به گونه ای که هزینه های کمتری را در بر داشته باشد. این کار با استفاده از واحدهای مختلف و درایه بود به گونه ای که هزینه های کمتری را در بر داشته باشد.

۳

B ₁	B ₂	B ₃	B ₄	B ₅	B ₆	B ₇	B ₈	B ₉	B ₁₀	B ₁₁	B ₁₂	B ₁₃	B ₁₄	B ₁₅	B ₁₆	B ₁₇	B ₁₈	B ₁₉	B ₂₀	B ₂₁	B ₂₂	B ₂₃	B ₂₄	B ₂₅	B ₂₆	B ₂₇	B ₂₈	B ₂₉	B ₃₀	B ₃₁	B ₃₂	B ₃₃	B ₃₄	B ₃₅	B ₃₆	B ₃₇	B ₃₈	B ₃₉	B ₄₀	B ₄₁	B ₄₂	B ₄₃	B ₄₄	B ₄₅	B ₄₆	B ₄₇	B ₄₈	B ₄₉	B ₅₀	B ₅₁	B ₅₂	B ₅₃	B ₅₄	B ₅₅	B ₅₆	B ₅₇	B ₅₈	B ₅₉	B ₆₀	B ₆₁	B ₆₂	B ₆₃	B ₆₄	B ₆₅	B ₆₆	B ₆₇	B ₆₈	B ₆₉	B ₇₀	B ₇₁	B ₇₂	B ₇₃	B ₇₄	B ₇₅	B ₇₆	B ₇₇	B ₇₈	B ₇₉	B ₈₀	B ₈₁	B ₈₂	B ₈₃	B ₈₄	B ₈₅	B ₈₆	B ₈₇	B ₈₈	B ₈₉	B ₉₀	B ₉₁	B ₉₂	B ₉₃	B ₉₄	B ₉₅	B ₉₆	B ₉₇	B ₉₈	B ₉₉	B ₁₀₀
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

ع-۲: کثیر القسط میں دو عدد متوالی یا قدر تکثیر (Resolution) برابر ہے، $\frac{1}{8} = 0.125$ ($= \text{ulp} = \text{unit of lower precision}$)

$$0.0625 = \pm \frac{1}{16} = \frac{\text{app. resolution}}{2} = \frac{\rho}{\text{other}} \quad \text{per } \epsilon$$

وقت عدم خدمت

$$2/1 \quad b_7 b_6 b_5 b_4 \frac{1}{3} b_2 b_1 b_0 = -4$$

$$N_{min} = 1000000 - 1/32$$

$$-\frac{1}{62} = -0.0161 \approx -0$$

0 - 000 000 000 - +

$0 = 00 \dots 01 = 1$
 $1 = 00 \dots 000001 = 1$

$$+\frac{1}{69} = 0.0145$$

$011 \triangle 11111 = 3$

$N_{\max} = 0.77 \times 10^6$

بالرکته $\sqrt{2}/2$

۱۵۰

$\tau_1 = \frac{1}{\omega} \arctan \left(\frac{\omega R C}{1} \right)$

والتالي

Handwritten signature

وہابیہ کی رائے میں حج کو فرض کیلئے

مفتی احمد رضا خان صاحب

میں نے اس کے لئے دعا کی ہے کہ وہ جلد صحت یاب ہو جائے۔

میزبانی است: $R_1 = R$

ADC $R_1, R_2 \Leftrightarrow R_1 - R_2$

R_5 و R_6 -

$R_3 +$

$\checkmark \Rightarrow$ Carry over 1, 1, 1, 0

Add R_2

Adc R₁

$$P_0 - P_1 - P_2 - \dots - P_n - R_1$$

$$\text{BB } R_1, R_2 \Leftrightarrow R_1 = R_2 - R_1$$

$$R_1, R_2 \quad \Leftarrow \quad \begin{matrix} SUB \\ ebb \end{matrix}$$

$R_3 \quad R_4 \rightarrow$

$\frac{13}{N}$ $N;$

Two

4-1 دستور Jump if greater (JG) مقدار مقایسه دو عدد و علامت را تعیین می کند.
 شرط پرش این است که $A > B$ یعنی $S=0$ و $Z=0$ $\Leftrightarrow A-B > 0$. از آن جا که اعداد علامت دار هستند پس کافی است بیت علامت مثبت با یکدیگر و سرریز با آن ها را اگر برعکس بیت سرریز یکسان باشد بیت علامت منفی باشد. لذا شرط مورد نیاز برای برآورد این عبارت $(\overline{SV} + SV) \cdot \overline{Z} = 1$ است.
 CMP A, B
 JG Addr.
 \Rightarrow JMP L2

L1: JO Addr.
 JMP Exit
 L2: JO Exit
 JMP Addr.
 !

Addr: Is greater
 Exit: Is not greater

2-4

$$\begin{array}{r} S_A \ a_{n-2} \dots a_0 \\ S_B \ b_{n-2} \dots b_0 \\ \hline S_d \ d_{n-2} \dots d_0 \end{array} + C_n$$

$$Overflow = \overline{S_A} \overline{S_B} S_d + S_A S_B \overline{S_d} \quad (1)$$

$$= C_{n-1} \oplus C_n$$

کافی است برآورد بیت سرریز و بیت علامت را واری می کنند

Test A
 JS L1 $S_A = 1?$
 Test B
 JS L2 $S_B = 1?$
 Test D
 JS Addr $S_D = 1?$
 JMP Exit
 \Rightarrow JO Addr

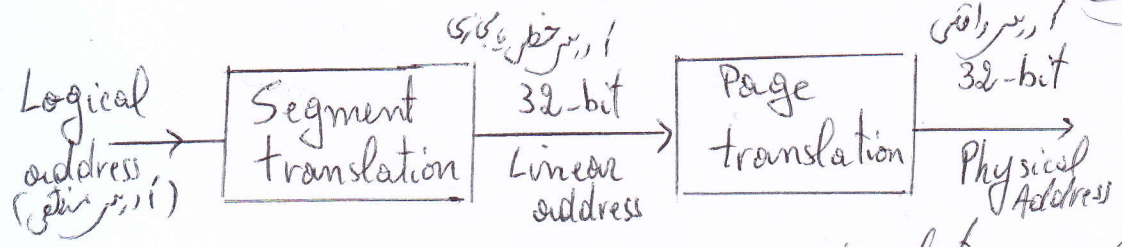
1-7

$$\begin{cases} CS: 028E \\ IP: 0040 \end{cases}$$

$$\Rightarrow Real (linear) Address = 028E0 + 0040 = 02920$$

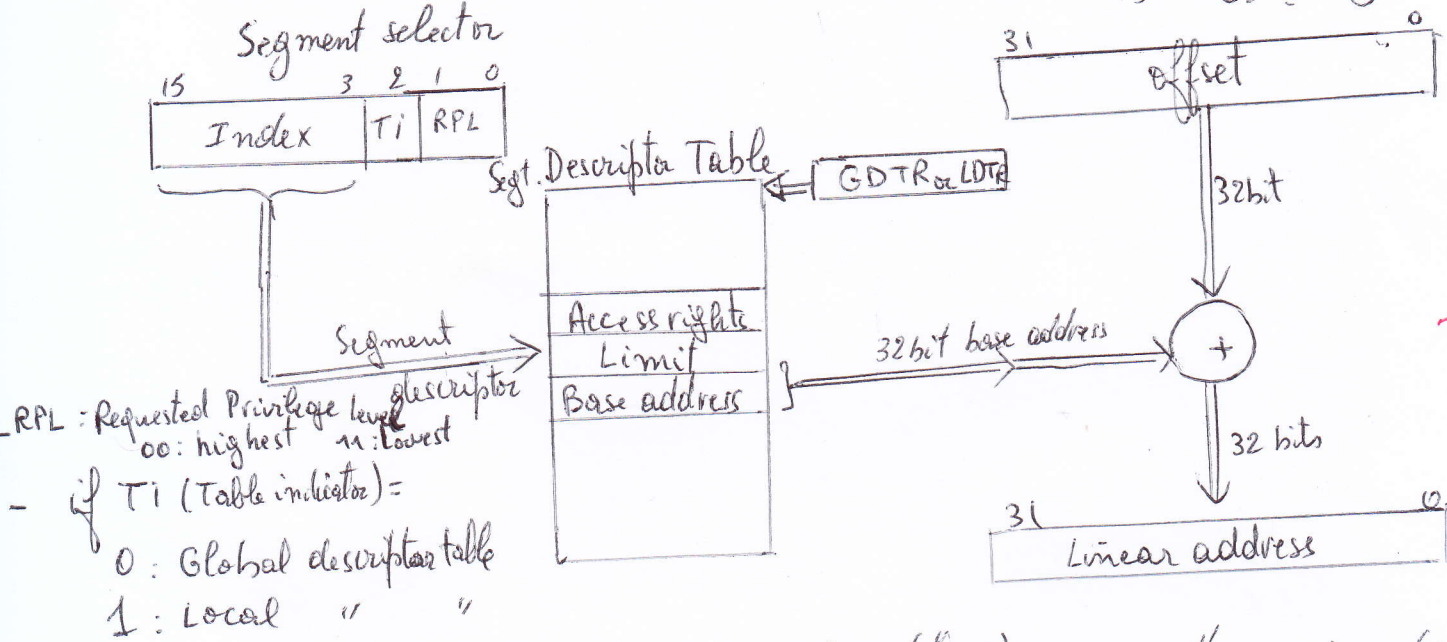
CS	0264	0263	0200	...
	IP	0000	0010	

آدرس منطقی از ترکیب آدرس فیزیکی



آدرس منطقی از ترکیب آدرس فیزیکی
 ۳۲ بیت Offset (EIP, ESP, EBP, ESI, EDI, EAX, EBX, ECX, EDX)
 ۳۲ بیت Segment selector (CS, SS, DS, ES, FS, GS)

واحد ترجه segment این آدرس منطقی را تبدیل به یک آدرس خطی (یا مبنی) می کند. برای این کار از یک جدول توصیف و یا آن segment استفاده می کند. در اداسه و یا به کارگیری یک جدول صفات و page directory آدرس مبنی را تبدیل به آدرس واقعی فیزیکی می کند.



- RPL: Requested Privilege level
00: highest 11: lowest

- TI (Table indicator) =
0: Global descriptor table
1: Local " "

GDTR: Global Descriptor Table Register (محتوای یکی در حافظه)
LDTR: Local " " (محتوای یکی در حافظه)

Segment Selector: CS, SS, ES, DS, FS, GS (16 بیت آدرس برای هر یک)
48 بیت از اطلاعات موجود در محل موردنظر که این آدرس را از جدول seg. در هیسین به نامت 4 به صورت مبنی و فیزیکی می کند.

محتوای جدول توصیف segment
seg حق دسترسی
اندازه seg و (R, W, RW, code, data)
آدرس مبنی را به آدرس واقعی تبدیل می کند.

۸-۶ یک تریج می فضا به نامت می شود اولاً به حسب RPL (سطح اولویت در فضا بین ۳ تا ۵)
مستحق گردد برای وظیفه (Task) مورد نظر حق دسترسی به فضا را دارد یا نه و ثانیاً به حسب جدول seg. Descriptor معلوم شود صریح دسترسی به فضا می باشد (دسترسی برای خواندن/نوشتن).

```

0 Data1 DW 2000h
2 Data2 DW 3000h
... Code
17 Les SI, Data1
1A mov di, offset Data2
1D mov bx, [SI]
1F mov cx, [di]
21 mov [SI], CX
23 mov [di], BX
Exit
END

```

۱-۹ این برنامه محتوا Data1 را، Data2 جایی می کند.
برای این کار آدرس Data1 در (ESI:SI) به یک Les قرار می دهد. سپس آدرس Data2 را به DI می گذارد و سپس به یک mov bx, [DI] جایی می کند.
۲-۹ دستور Les mov di, offset Data2 را می بینیم. این دستور ۳ عدد و یا به نامت آدرس را به DI می گذارد و DI را به BX می گذارد. (Les BX, EDI)
۳-۹ کافی است در آدرس 1F که 3 بیت یعنی CC hex را قرار دهیم و در آنجا 3 بیت را به DI می گذارد. (push F)
۹-۱ کافی است بیت T را به 1 می گذاریم و در آنجا 3 بیت را به DI می گذارد. (pop F)