

1. Uzupełnij kod tak żeby wrzucić liczbę na stos koprocessora liczbę 30.

```
push dword ptr 30
```

```
....
```

```
add esp 4
```

Odp.: `fild dword ptr [esp]`

2. Czemu architektura von Neumana jest niedokładna na potrzeby współczesnego programowania?

Odp.: W tej architekturze rozkazy i dane są przechowywane w pamięci w sposób identyczny, czyli kod nie pozwala odróżnić instrukcji od danych. Ta architektura to taki typowy komputer SISD - do procesora dochodzi pojedynczy strumień instrukcji, które operują na pojedynczym strumieniu danych, podczas gdy współczesne komputery to raczej są SIMD i MIMD.

3. Opisać algorytm LRU

Odp.: LRU (ang. Least Recently Used) - algorytm stronicowania. LRU polega na zastępowaniu w cache'u jako pierwszej najdawniej używanej strony. Wymaga to informacji o tym, kiedy poszczególne strony były używane, co jest kosztowne, jeśli chce się mieć pewność, że wyrzuci się rzeczywiście stronę używaną najdawniej. Do pamiętania czasu użycia stosuje się różne algorytmy, z których dwa zostały pokrótce opisane poniżej:

- Liczniki

Do każdej pozycji w tablicy stron dołączany jest rejestr czasu użycia, do procesora zaś dodaje się zegar logiczny lub licznik. Wskazania zegara są zwiększane wraz z każdym odniesieniem do pamięci. Ilekroć występuje odniesienie do pamięci, tylekroć zawartość rejestru zegara jest kopiowana do rejestru czasu użycia należącego do danej strony w tablicy stron.

- Stos

Przy każdym odwołaniu do strony jej numer wyjmujemy ze stosu i umieszczamy na szczycie - najlepsza implementacja to dwukierunkowa lista ze wskaźnikami do czoła i do końca. Wystarczy najwyżej 6 zmian wskaźników. Nie jest potrzebne przeszukiwanie listy.

4. Ciąg mikroinstrukcji dla:

```
add bl,bh
```

Odp.:

5. Instrukcja ma dodać "1" do liczby stałoprzecinkowej zapisanej w rejestrach EDI:EDX. Uzupełnij komendę:

ADD EDX, 1

ADC EDI, ...

Odp.: Ja bym napisał *ADD EDI, 0* ; *adc* dodaje do rejestru przeniesienie z poprzedniej operacji, więc jeśli w *ADD EDX, 1* nastąpiło przeniesienie to zostało ono zapisane do CF, a CF zostanie dodany do EDI poprzez rozkaz *ADC*.

6. Dlaczego w systemach z dużą pamięcią operacyjną nie stosuje się adresowania asocjacyjnego?

Odp.: Adresowanie asocjacyjne jest wolne, co w połączeniu z dużą pamięcią działałoby tragicznie. W adresowaniu asocjacyjnym za rozróżnianie bloków pamięci odpowiada komparator, który musi liniowo porównywać etykiety. Czym ich będzie więcej tym wolniej będzie to działać.

7. Instrukcja MOV EDX, dword PTR BH jest błędna. Jak prawidłowo przesłać BH do EDI?

Odp.: *MOVZX EDX, BH*

lub *MOVSX EDX, BH*

w zależności od tego, na jakich liczbach operujemy (ze znakiem/bez znaku)

8. Czym różni się rozkaz jmp dword ptr [esp] od rozkazu ret?

Odp.: Rozkaz skoku *jmp* ustawia EIP na adres, który znajduje się na wierzchołku stosu. *Ret* robi to samo, ale dodatkowo zrzuca ten adres ze stosu.

9. Do czego służy tablica przerwań deskryptorów?

Odp.: Tablica ta zawiera adresy procedur obsługujących dane przerwanie. Jej zadaniem jest skojarzenie danej procedury z IRQ przerwania.

10. Podać wynik działania X/Y lub X-Y w postaci liczby dziesiętnej. Liczby X i Y są zmiennoprzecinkowe:

X = 1 10000001 10100010100001001010100

Y = 1 10000010 10100010100001001010100

Odp.: $X = (1+m) \cdot 2^{(129-127)} = (1+m) \cdot 2^2$

$Y = (1+m) \cdot 2^{(130-127)} = (1+m) \cdot 2^3$

Wyrażenie $(1+m)$ jest równe dla x i y . Zatem $X/Y = 4/8 = 0.5$, $X-Y = 4-8 = -4$

11. Pamięć podręczna dwukanałowa, Adresowanie 36bitowe, etykieta 16 bitów, offset 8bitów, jaki rozmiar pamięci.

Odp.:

| etykieta | nr linii | offset |

|16 bitów| 12 bitów | 8 bitów|

2 kanały,

Rozmiar = $2^8 * 2^{12} * 2 = 512 \text{ kB}$

12. Na czym polega przewidywanie statyczne rozkazów.

Odp.: Skoki przewidywane przez kompilator lub programistę:

Kompilator języka wysokiego poziomu lub programista piszący program w assemblerze zna prawdopodobieństwo wykonania skoku warunkowego.

Przewidywanie przez procesor:

z analizy własności programów wynika, że ponad 60% skoków warunkowych w tył jest realizowanych, a skoków w przód - nierealizowanych, skoki warunkowe są zapisywane jako względne - bit znaku przemieszczenia może zostać użyty przez procesor jako znacznik prawdopodobieństwa skoku.

13. Podać hipotetyczny ciąg mikroinstukcji dla rozkazu

pop ebx

Odp.:

14. W jaki sposób działają układy DMA w komputerach?

Odp.: Direct Memory Access, DMA (z ang. bezpośredni dostęp do pamięci) – technika, w której sprzęt komputerowy podłączony do płyty głównej, np. karta graficzna, karta dźwiękowa, karta sieciowa czy kontroler dysku twardego, mogą korzystać z pamięci operacyjnej RAM lub portów we-wy, pomijając przy tym CPU. Wymaga to niewielkiej współpracy ze strony procesora, który musi zaprogramować kontroler DMA do wykonania odpowiedniego transferu danych, a następnie na czas przesyłania danych zwolnić magistralę systemową (przejsć w stan wysokiej impedancji). Natomiast sam transfer danych jest już zadaniem wyłącznie kontrolera DMA. Realizacja cykli DMA może być przejmowana przez dedykowany układ cyfrowy, tak jak np. w komputerach PC, lub być realizowana programowo przez dane urządzenie. DMA ma za zadanie odciążyć procesor główny od przesyłania danych.

15. Opisać budowę pamięci SRAM.

Odp.: Jest to pamięć, która ze względu na swoją budowę nie wymaga odświeżania danych. Ma ona stałe zasilanie, zbudowana jest z tranzystorów tworzących przerzutniki. Jest to układ drogi i prądożerny. Jest jednak 7 razy szybsza od DRAM. Pamięć ta stosowana jest w pamięci cache procesora, gdyż występuje tam pamięć niewielkich rozmiarów.

16. Dlaczego asembler wywali błąd przy tej instrukcji:

```
sub [esi+edi], 1
```

Odp.: Należy dopisać rozmiar danych, na jakich ma zostać wykonany rozkaz np.

```
SUB [ESI+EDI], BYTE PTR 1
```

17. Czym się różni pamięć DRAM od SDRAM?

Odp.: SDRAM to pamięć synchroniczna, która jest odświeżana zgodnie z cyklem zegara podczas gdy pamięć ta nie jest aktualnie używana. Pamięć DRAM jest odświeżana w dowolnym czasie, "zatrzymując" na chwilę pracę procesora, w tym przypadku czas procesora jest marnowany.

18. Dlaczego w systemach komputerowych, gdzie jest wiele proców i wspólna magistrala danych stosuje się zapis z opóźnieniem, a nie zapis przez?

Odp.:

19. Opisać ciąg mikroinstrukcji dla rozkazu loop.

Odp.:

20. Dlaczego wyrównywanie danych jest ważne?

Odp.: Prawdopodobnie chodzi o przetwarzanie potokowe, by odczytywać wszystko w danym cyklu, co pozwoli na szybkie przetwarzanie danych.

21. Jakie znaczenie ma flaga IF w obsłudze przerwań?

Odp.: Znacznik IF służy do włączania/wyłączania obsługi przerwań. Zeruje się go podczas wpisywania adresu procedury, gdyż może się zdarzyć, że w tym czasie przyjdzie przerwanie i dojdzie do tego, że część adresu będzie poprawna, a druga się nie wpisze, co z kolei spowoduje, że adres będzie błędny i losowy.

22. Instrukcja zapisana w AT&T zamienić na intelowski: movl \$4, %esi

Odp.: MOV ESI, 4

23. Jaki będzie stan EIP po wykonaniu ciągu instrukcji:

```
mov ebx, 724
```

xchg [esp], ebx

ret

Odp.: EIP = 724

24. Dlaczego taka instrukcja zawsze spowoduje błąd:

div edx

(wskazówka: div dzieli liczbę 64-bitową zawartą w EDX:EAX)

Dzieląc liczbę EDX:EAX przez EDX masz tylko 2 opcje:

- EDX = 0 - no i divide by 0 więc leci wyjątek
- EDX != 0 - no i liczba nie mieści się w EAX więc też leci wyjątek

25. Opisać organizację (hierarchię?) pamięci podręcznej w procesorach Intel Core 7.

Wszystkie procesory z rodziny Core i7 mają taką samą ilość pamięci cache:

- po 32KB pamięci instrukcyjnej L1 i 32KB pamięci danych L1 na każdy z rdzeni
- po 256KB współdzielonej pamięci instrukcyjnej/danych L2 na każdy z rdzeni
- 8MB współdzielonej pamięci instrukcyjnej/danych L3 wspólnej dla wszystkich rdzeni

26. Zaznaczyć na diagramie wartość/przedział CL (CAS latency) - aktywacja narastającym zboczem.

Odp.:

27. W wyniku dołożenia do 6 procesorów jednego uzyskano przyspieszenie 5%, jaki procent operacji wykonuje się równolegle?

Odp.: $S(n) = \frac{s + p}{s + \frac{p}{n}}$

$$S(7) = S(6) * 1.05$$

$$s = 1 - p$$

$$\frac{1-p+p}{1-p+\frac{p}{7}} = \frac{1-p+p}{1-p+\frac{p}{6}} \cdot \frac{21}{20}$$

$$\frac{1}{1-p+\frac{p}{7}} = \frac{1}{1-p+\frac{p}{6}} \cdot \frac{21}{20}$$

$$\frac{1}{1-\frac{6}{7}p} = \frac{1}{1-\frac{5}{6}p} \cdot \frac{21}{20}$$

$$\frac{1}{1-\frac{6}{7}p} = \frac{21}{20-\frac{100}{6}p}$$

$$21 - \frac{126}{7}p = 20 - \frac{100}{6}p$$

$$-\frac{126}{7}p + \frac{100}{6}p = -1$$

$$-\frac{756}{42}p + \frac{700}{42}p = -1$$

$$-\frac{56}{42}p = -1 \quad p = \frac{42}{56} = \text{ok. } 75\%$$

28. Opisz, czym są hazardy zasobów i podaj sposoby na ich minimalizację.

Odp.:

Problemy z przetwarzaniem potokowym – hazardy:

- Hazardy strukturalne : konflikty dostępu do zasobu – pamięci, rejestru, ALU.
- Hazardy danych: Jeden rozkaz może wykorzystywać dane będące produktem wykonywania innego rozkazu – czyli musi poczekać.
- Hazardy sterowania: Skoki warunkowe zmieniają kolejność wykonywania instrukcji – w konsekwencji trzeba czyścić cały potok z załadowanych już rozkazów.

Hazardy – przeciwdziałanie:

- Optymalizacja kodu na etapie kompilacji (zmiana kolejności instrukcji, które można szeregować wpływu na rezultaty kodu)
- Przewidywania statystyczne i dynamiczne (dotyczy prawdopodobieństwa wystąpienia skoku)
- Bufor pętli

29. Przedstaw sposób wyświetlania grafiki w trybie 13H.

Odp.:

13H jest prostym trybem graficznym używanym w rzeczywistym trybie pracy procesora, w środowisku DOS. Jego rozdzielczość to 320 na 200 pikseli czyli 64000B. Każdy piksel to jeden bajt w pamięci który określa jego kolor. Adresy w pamięci począwszy od A0000H zawierają informację o danym pikselu.

30. Cemu w mikroarchitekturze stosuje się zmiany nazwy rejestrów (register renaming)?

Odp.:

Przemianowanie rejestrów - technika w inżynierii komputerowej używana w celu uniknięcia niepotrzebnego szeregowego wykonania instrukcji narzuconego przez wykorzystanie tych samych rejestrów procesora przez następujące po sobie instrukcje.

Kiedy więcej niż jedna instrukcja odwołuje się do określonego operandu w celu odczytu, bądź zapisu, wykonanie tych instrukcji w innej kolejności niż wynika to z oryginalnego porządku programu skutkuje konfliktem danych:

Odczyt po zapisie (RAW)

Instrukcja odczytująca zawartość rejestru musi wykonać się po instrukcji zapisującej występującej bezpośrednio przed nią w porządku programu, a nie po innej instrukcji zapisującej ten rejestr.

Zapis po zapisie (WAW)

Zawartość rejestru po sekwencji instrukcji piszących do niego musi wynikać z ostatniej instrukcji według porządku programu.

Zapis po odczycie (WAR)

Instrukcja zapisu nie może wykonać się po instrukcji odczytu z tego samego rejestru, gdyż spowoduje to odczytanie błędnej wartości.

31. Wyjaśnić w jakich okolicznościach pojawia się wyjątek koprocesora niedozwolona operacja.

Odp.:

Ten wyjątek jest generowany gdy niemożliwe jest żadne inne działanie; wygeneruje go np. próba pierwiastkowania liczby ujemnej, odwołanie się do pustego rejestru stosu koprocesora. Możliwe jest zamaskowanie wyjątku poprzez ustawienie na 1 bitów IM.

32. Co robi MOVSV i ten MOVZX.

Odp.: Wpisuje dane z rejestru mniejszego do większego. Różnica pomiędzy tymi rozkazami to ze znakiem/bez znaku.

33. Do 9 procesorów dodano 1 i uzyskany przyspieszenie 5%. Jaki procent obliczeń jest wykonywany w sposób równoległy?

Odp.: Patrz zadanie 27.

34. Pokazany kawałek pamięci, jaka wartość zostanie wpisana do rejestru przy założeniu little endian?

Odp.: Łatwe.

35. Czym jest lane w komputerach/procesorach wektorowych i jaki jest tego odpowiednik w GPU?

Odp.: