КАРХ

**1. Какво е характерно за регистрите с общо предназначение при 64-битов режим?**

Тези регистри се използват за съхранението на операндите на логическите и аритметични операции. Те също така могат да бъдат използвани за операнди на изчисленията на адрес.

**2.Каква е ролята на завършващата част?**Завършващата част се грижи резултатът от изчисленията да съответства на реда на постъпване на инструкции и че състоянието на системата се обновява правилно.

**3. Защитеният режим (protected mode).**

Той се характеризира с апаратно предотвратяване на достъпа на дадена програма извън границите на заделената и памет.

**4. Обяснете механизма за обновяване на страница от диска**

Мениджърът на паметта използва информацията от таблиците за да намери неизползвана страница физическа памет (такава, към която скоро не е имало обръщения) и я записва в резервирана област на диска, наречена swap файл. После уведомява централния процесор за необходимостта да прочете изискваната страница от по-бавната памет (файл на диска или swap файл). Накрая MMU установява съответствието между виртуалната и физическата страница и обновява таблицата с тази информация.

5. **Опишете последователността от стъпки, която се изпълнява при прекъсване при IA32**

Аварийният цикъл се състои в следното:

1. Запазва се векторното състояние на текущата програма;   
2. Определя се видът на аварийното събитие или чрез сигнал от линиите за прекъсване, или като код за линиите за данни към процесора;   
3. По кода на вида прекъсване се извлича от оперативната памет нов вектор на състоянието, т.е. се прави зареждане с ново съдържание на централния процесор, включително и program counter-а. Старите регистри отиват в друга област – старо състояние. Новото и старото състояние са области в операционната памет;   
4. Преход към нормален цикъл. В резултат на новото съдържание процесорът тръгва от нова команда.

6. **Обяснете групите, на които се делят изключенията**

1) Грешки (*faults*). Грешките се откриват и обслужват непосредствено преди следващата команда. Най-често се явяват в системата на виртуалната памет, когато е заявен адрес към несъществуваща страница или сегмент. Съобщението подготвя процесора за рестарт на командата, генерирала грешката, т.е. адресът за връщане от обслужване на прекъсването представлява старото съдържание на програмния брояч. В същото време операционната система търси нужната страница или сегмент.

2)Капани (*traps*). Съобщението за капан, в който е попаднал хода на изчислителния процес, се генерира непосредствено в края на текущата и преди следващата команда. Капаните това са контролни точки, които могат да бъдат залагани умишлено от потребителя, с цел контрол на изчислителния процес, обикновено при тестване на софтуера, чрез вмъкване в кода на програмата на еднобайтова команда INT 3.

3) Повреди (аварийни прекъсвания) (*aborts*). При такива ситуации не винаги се разполага с адреса на командата, по време на която е настъпило изключението. По тази причина не винаги е възможен рестарт на командата и следователно на хода на изчислителния процес. Типични изключения от този вид са различни апаратни грешки, откривани от контролиращи логически схеми, или при противоречиви (недопустими) стойности в системните таблици.

**7. GDTR (****Global Descriptor Table Register)** – този регистър сочи към началния адрес на глобална дескрипторна таблица (Global Description Table), която се използва при сегментирания модел на адресация.

**8. Какво е предназначението на сегментните селектори**

Сегментните селектори са 16 битови идентификатори на сегмента. Старшите 13 бита се използват за определяне на отместването в глобалната или локалната дескрипторни таблици. Трите младши бита имат по-специфично предназначение.При сегментацията адресът се получава от 16-битов сегментен регистър и 32-битово отместване.

9. **Как работи изпълняващата част?**Изпълняващата част пренарежда микрооперациите по такъв начин, че тези чиито операнди са готови (и има налични изчислителни ресурси) да се изпълнят възможно по - скоро. Изпълняващата част може да обработи (issue) няколко микрооперации за един цикъл.

10. **Прилики и разлики между прекъсване и изключение**

Иключението възниква в тялото на текущата програма, където работи процесорът, и се обработва в нея. Обикновено това събитие е породено от невъзможността на процесора да извърши текущата команда. Например в резултат на прехода program counter (програмният брояч) да сочи някъде извън реалната памет или да се окаже, че командата е наред, но има адрес на една от данните, която сочи извън паметта. Други такива команди са: опит за делене на 0; умножение на две големи числа, при което битовете не достигат; опит на програмата да пише в област, в която няма право и др. Прекъсванията възникват винаги извън тялото на текущия процес, т.е. възникват от входно-изходните устройства, схемите на централния процесор или от други схеми. Тези събития се обработват от операционната система, извън тялото на текущия процес. И в двата случая обработката на аварийната ситуация се извършва първо от апаратурата, като централният процесор преминава в авариен цикъл на изпълнение, и след това от софтуерен модул, който обработва данните с цел да намали щетите от възникналото събитие.

11. **Как се осъществява превключване между задачите**

Операционната система има модул, наречен loader (зареждащ модул), който, получавайки съответна команда, извършва зареждане на задача за предстоящо изпълнение, като при това зареждане в оперативната памет за тази задача се създава споменатия вече (TSS). Освен този модул, при управлението на задачите участва и още един – планиращ модул (модул за планиране), който решава коя от задачите, заредени в оперативната памет, да активира. Той работи на следния принцип:  
1) Издава команда към процесора за задействането му след даден квант време.  
2) Обхожда таблицата на задачите, взима първата, която е в режим на готовност и я стартира, след което продължава по следния начин – записва състоянието на предишната и започва да работи по новата задача.  
3) След като изтече квантът време, процесорът е бил задействан от таймер и този вид прекъсване води до активиране на планиращия модул, който отново издава команда за задействане на процесора и отново сканира таблицата. Има статистическа информация, която пренарежда таблицата, така че да не се позволи една и съща задача да се изпълнява многократно постоянно.

12. **Какво представляват и какви са предимствата на пакетираните SIMD типове данни**

SIMD типовете данни са MMX и XMM. ММХ типът е разработен, за да ускори мултимедийните програми и програмите за комуникация като включват нови инструкции и типове данни които позволяват на програмите да достигат ново ниво на производителност. ММХ типът е разработен като множество от integer инструкции които могат да бъдат приложени към нуждите на голямо разнообразие от мултимедия.

13. **Каква информация се съдържа в IDT?**

В Protected mode за всяка програмна част, която се записва някъде, се създава един дескриптор, който представлява таблица на прекъсванията IDT (Interrupt Descriptor Table) и за нея има един регистър, който сочи началото на тази таблица. Когато дойде код на прекъсване, той се разглежда като код на отместване в таблицата, откъдето се определя началото на съответния програмен модул.

14. **По какво се различават изключенията от тип faults и abort**

За faults се знае коя инструкция ги е причинила и програмата или процесът могат да бъдат рестартирани без загуба, докато при

Aborts това може да не е ясно и следва прекратява на програмата или процесът.

Изключенията от тип грешки (fault)като цяло могат да бъдат коригирани и за това след като се случат и бъде изпълнена обработващата програма, се стартира отново същата програма (която е генерирала грешката), като се счита, че грешката вече е коригирана (от обработващата програма).

15. **Каква е функцията на челната част (front-end)**

Челната част (Front End) на вътрешната архитектура на IA-32 е един от основните компоненти на процесорите Intel, както може да се види на фигурата. Тя съдържа кеша за проследяване, който е кеш за инструкции. При липса в кеша за проследяване в непосредствена близост до кеша за проследяване е разположен ROM c микрокода, който съхранява микрооперациите на комплексните инструкции. За комплексните инструкции кешът за проследяване изпраща указател към ROM, по който се извлича съответната последователност от микрооперации, имплементиращи тази инструкция.

16.