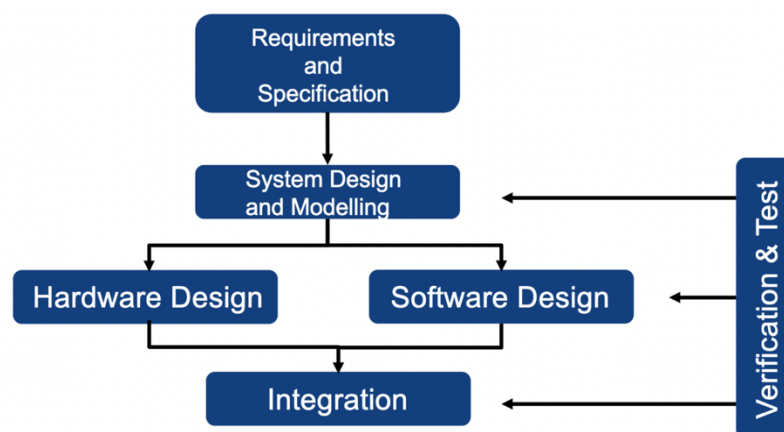


Embedded system, Exams questions

1. **Hur brukar ett inbyggt system definieras?**
 - a. Det är ett system som integrerar den centrala processorenheten och dators minne, samt in- och utgående kringutrustning, som har den specifika funktionen av ett större mekaniskt eller elektroniskt system.
2. Vad är det för skillnad mellan en mikroprocessor och mikrokontroller?
 - a. Mikroprocessor är ett chip som är programmerbart med flera komponenter som är kärnan till hela datorsystemet som består av ALU, Control Unit, instruktionsdekodare och Register Array. Dessutom är mikroprocessor en integrerad krets utan RAM, ROM eller I/O pinnar. Därmed kräver mikroprocessor en extern buss för gränssnitt till kringutrustning till RAM, ROM, Analog och Digital IO och seriella portar. Mikroprocessor är en masslagringsenhet med bara ett chip som är inbäddade i flertal applikationer som trafikljusstyrning och temperaturkontroll.
 - b. Å andra sida är mikrokontroller kärnan av det inbyggda systemet som är förening av mikroprocessortekniken. Tillskillnad från mikroprocessor så är mikrokontroller mycket kraftfullare enheter som innehåller ROM, I/O pinnar i samma chip. Dessutom är mikrokontroller utformat för att göra specifika uppdrag som att exekvera program för att styra andra enheter. Mikrokontroller användas vidare i elektriska och elektroniska kretsar och automatiska styrda enheter som spelmaskin och solladdare.
3. **Ge exempel på utmaningar vi har när det gäller att designa ett inbyggt system.**
 - a. Funktionskrav: som beskriver vad systemet måste göra (produkttegenskaper)
 - b. Icke-funktionella krav: som är kvalitetsrelaterade krav
 - i. Prestandan
 - ii. Pålitlighet
 - iii. Testbarhet
 - c. Arkitektoniska krav: som beskriver hur olika delsystem ansluter till varandra och namnkonventioner
 - d. Price, low cost • Light weight • Reliability • Low power • Portable • Complexity
4. **Hur ser normalt processen ut för att designa ett inbyggt system?**



5. I STM32 finns något som kallas "ARM Cortex-M4". Vad innebär det?
 - a. ARM Cortex-M4 är en mycket effektiv inbäddad processor som är utvecklad för digital signalkontroll som begär en effektiv, lättanvänd blandning av kontroll- och signalbehandlingsmöjligheter.
6. I Cortex CPU finns en "3-stage prefetch". Förklara principen!
 - a. Det betyder att det finns tre steg som en data går igenom:
 - i. The Fetch stage: när data returneras från instruktionsminnet.
 - ii. The Decode stage: generering av LSU-adress med hjälp av forwarded registerportar, omedelbar offset eller forwarded av LR-registergren.
 - iii. The Execute stage: en enkel pipeline med flercykelstall, LSU-adress/datapipelining till AHB-gränssnitt.
7. Förklara vad som menas med att multiple interrupts will be tail-chained.
 - a. Om ett interrupt pågår och ett annat interrupt av lägre prioritet sker så kommer den exekveras direkt efter det första utan att återställa hela kontexten förrän båda interrupts är klara. Därmed sparas några minnes cyklar och mindre tid samt energi utnyttjas.
8. Vad är syftet med "Cortex Microcontroller Software Interface Standard" (CMSIS)?
 - a. CMSIS bidrar till att förenkla programvara återanvändning, minska inlärningskurvan för mikrokontroller utvecklare, påskynda projekt bygga och felsöka.
9. Det finns tre olika "boot modes", dvs hur man startar exekvering av program. Vilka?
 - a. Main Flash memory.
 - b. System memory.
 - c. Embedded SRAM.
10. Det finns möjlighet att starta en boot loader. Vad kan den användas till?
 - a. Det är en speciell operativsystemprogramvara som laddar och startar starttidsuppgifterna och processerna för ett operativsystem eller datorsystemet. Det gör det möjligt att ladda operativsystemet i datorns minne när en dator startas eller startar upp.
11. Vad innebär det att en variabel är deklarerad som static?
 - a. En statisk variabel initieras endast en gång. Kompilatorn kvarstår med variabeln till slutet av programmet. Statiska variabler kan definieras inuti eller utanför funktionen. De är lokala i blocket. Standardvärdet för statiska variabler är noll och de är livande till programmet körs.
12. När kan det vara lämpligt att använda malloc?
 - a. Malloc används för att dynamiskt allokera minnesutrymme på heapen. Det kan vara lämpligt att använda när man använder sig av pointers då c inte allokerar minne automatiskt.
13. Hur gör man för att variabler ska hamna i heap? När kan det vara lämpligt att reservera plats för variabler i heap?
 - a. Allt minne som vi allokerar genom dynamiska minneshantering (malloc och free) kommer att hamna i den delen av datorns minne som vi kallar heap. Heapen tar upp hela datorns RAM.
14. Förklara vad som menas med pekare och ge ett exempel på när det är nödvändigt att använda pekare.
 - a. pekaren är en speciell variabel som användas för att lagra adress av en annan variabel. En pekare kan också användas för att referera till en annan

pekarkfunktion. Syftet med pekare är att spara minnesutrymme och uppnå snabbare exekveringstid.

15. Varför deklareras vissa variabler som volatile? Ge exempel på när det kan vara lämpligt att deklarera så.

- I/O-enheterna är inte riktiga minnen därför kan innehållet av deklarationer ändras av sig själv när programmet körs. Med hjälp av volatile kan kompilatorn förstå att dessa deklarationen måste behålla sina innehåll och ändra inte på den. Detta görs vid programmering av en processor. Ett bra exempel en processor som har cache-minne så det är viktigt att man läser klock-kretsen direkt och inte några äldre cachade tider.

16. Vad betyder förkortningen API? Förklara vad som menas?

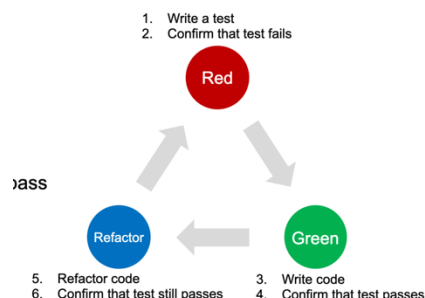
- API står för application programming interface, vilket är en uppsättning definitioner och protokoll för att bygga och integrera applikationsprogramvara. Dessutom hjälper API två program att kommunicera med varandra genom att erbjuda rätta verktyg och funktioner.

17. Gör en jämförelse mellan linjär spänningsregulator och switchad spänningsregulator. Vilka för- och nackdelar finns det vid användning i ett inbyggt system.

Linjär spänningsregulator	Switchad spänningsregulator
<ul style="list-style-type: none">• Billig och tar mindre utrymme på kretskort.• Är effektiva i ett litet antal fall som en 5V mikroprocessor i viloläge.• Konstant utspänning även om inspänning varierar.• Drop out Voltage (U-ut) är lite mindre än drop in Voltage (U-in).• Bra för low powered devices.	<ul style="list-style-type: none">• Dyrare.• Är bäst när strömeffektiviteten är kritisk.• Avger mer ljud.• Drop out Voltage (U-ut) kan vara högre än drop in Voltage (u-in) omvänd polaritet.• Mer komplicerade.

18. Redogör för hur programmering enligt TDD (Test Driven Development) går till.

- Lägg till ett test.
- Kör testet, som kommer att misslycka.
- Börjar skriva koden.
- Kör testet igen, ändra koden tills den klarar testet utan något fel.
- Refaktor koden och rensa upp koden.
- Fortsätta med nästa test.



19. Vad menas med ett VCS (Version Control System). Motivera varför det kan vara nödvändigt att använda ett VCS.
- Det är en specialiserad programvara vars primära mål är att hantera ändringar av kodbasen över tid. Huvudsyftet med VCS är att kunna återkalla specifika versioner senare. Ett exempel på VCS är Git. En VCS gör dessutom möjligt för utvecklare att snabbt återställa programvaran till det senast kända tillståndet i händelse av ett fel eller en bugg.
20. Rita ett kopplingsschema för en lämplig koppling för att ansluta en lysdiod till en utgång på mikrokontroller.
21. Beskriv vad det innebär att en utgång är konfigurerad som push-pull respektive open-drain.
- Open drain: man ansluter en transistor (MOSFET) till låg och inget annat kopplas mer.
 - Push-pull: man ansluter två transistorer (MOSFET) den ena till hög och den andra till låg.
 - När ett magnetfält slår på sensorn för open drain:en, leder MOSFET:en ström, vilket gör att ström sjunker genom pull-up-motståndet till jord.
22. Ge exempel på när det kan vara lämpligt att konfigurera en ingång som pull-up respektive floating.
- En pull-up anslutning kan vara mellan en tryckknapp och en ingång.
 - en float kan vara bra för input av analoga komponenter
23. Beskriv två metoder att avstudsas en kontakt med hårdvara.
- SR-latch
 - När kretsen föreställas i utgångsdelen av switch kommer den att behålla ingångens spänningsnivå som utgångstillstånd. Därför latching till ingången, när förändring i tillstånd införs. Vidare är denna metod användbar, men ökar den enkla kretsens svårhanterlighet samt den är dyrare än RC-net.
 - RC-net
 - Denna krets involverar kombinationen av ett motstånd och en kondensatorkrets för att fungera som ett filter för att jämna ut utgångsfelet för switchen.
24. Beskriv hur avstudsning av en kontakt kan göras i programvara.
- Man kan eliminera switchens studsande tillståndseffekt med hjälp av olika algoritmer och filter. Programmeraren kan designa en algoritm med användning av skiftregister och räknare så att den registrerar switchens tillstånd efter en fördröjning. En annan metod är att använda filteralgoritm på den samplade ingången från switch och bestämma switchens tillstånd baserat på utsignalen från ett sådant digitalt filter. Allt detta kan göra programvaran något ineffektiv, vilket ökar prestandafördröjningen om den inte implementeras korrekt.
25. Vad betyder MEMS och ge ett exempel på några sensorer som kan vara utfärda med MEMS.
- Det betyder micro-electro-mechanical-system och det definieras som miniatyriserade mekaniska och elektromekaniska element (d.v.s. enheter och strukturer) som tillverkas med hjälp av mikrotillverkningstekniker. Exempel på sensor som är utfärda med MEMS är Accelerometer och Gyroscope.

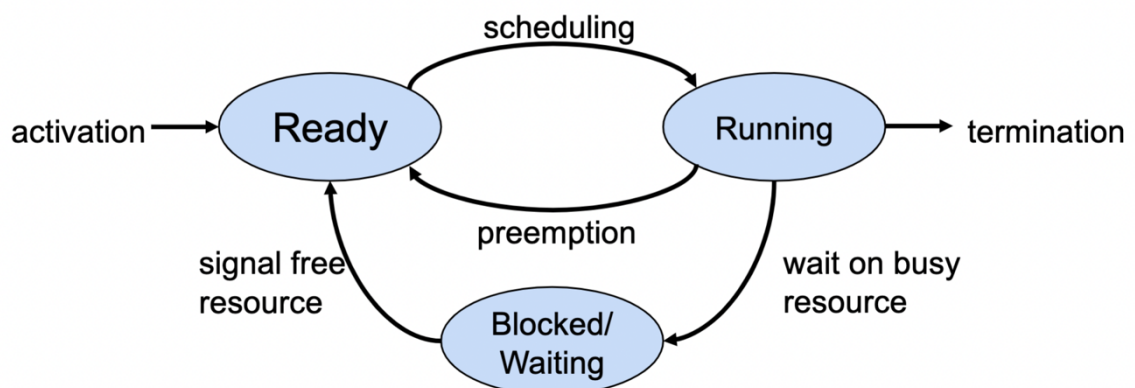
26. En A/D-omvandlare har 10 bitar. Hur många spänningsnivåer motsvarar det? Antag att vi har 3V referensspänning, vilken upplösning i spänning ger det med 10 bitars omvandlare.
- Om A/D-omvandlare har 10 bitar så spänningsnivåer kommer att vara 2^{10} vilket är 1024 spänningsnivåer. Om vi har 3V referensspänning som upplösning är $(\text{spänningen}/2^{\text{antal bitar}}) = 3V/2^{10} = 0,00029 \text{ V}$.
27. En ljudsignal var frekvensområde begränsar till 10kHz skall samplas för att läsa in i en mikrokontroll och signalbehandlas. Vilken lägsta samplingshastighet behöver vi?
- Nyquist Theorem säger att den lägsta samplingshastighet måste vara $2 \cdot f_{\text{max}}$. Detta gör att Nyquist rate blir 20 kHz.
28. Vi vill styra en ström på 50 mA från en digital utgång på mikrokontrollen. Ge exempel på en koppling som kan lösa detta problem.
- Se H-Bridge
29. Vi vill i ett inbyggt system alstra en rörelse med hjälp av en motor. Vi har att välja mellan stegmotor och en likströmsmotor. Jämför de båda alternativen och ange för- respektive nackdelar med respektive val.
- Stegmotor behöver ingen återkoppling, den styrs digital, den är dyr och responstid är låg.
 - Likströmsmotor kräver en återkoppling för att kunna styra läget, hastigheten styrs av spänningen, den styrs av PWM och den är billig.
30. Antag att vi i föregående uppgift valt motor. Föreslå hur vi automatiskt kan hitta en startposition för rörelsen.
- Med hjälp av en Encoder och tillhörande mjukvara kan en precis rotations position läsas av. Detta är inte nödvändigt för en stegmotor då de jobbar i exakta steg. alltså om man kan hålla koll på vilka steg som har tagits behövs inte rotationspositionen läsas av.
31. Förklara hur vi med timerfunktionen i mikrokontrollen kan styra varvtalet för en likströmsmotor.
- Timer i mikrokontroller kan användas för PWM segama, om spänningen max blir 5V och vi har en duty cycle 100%, så får vi 5V, om duty cycle är 50% då får vi 2,5V. Duty cycle beskriver hur länge är hög.
32. Föreslå en elektronikkoppling för att växla rotationsriktning på en likströmsmotor.
- krävs högre ström än vad mikrokontrollen kan ge. Externa transistorer kan användas genom mikrokontrollens pinnar.
33. En likströmsmotor är en form av induktiv last. Vad är risken med att bryta strömmen till en induktiv last och hur kan vi eliminera den risken.
- När strömförsörjningen stängs av och inte längre levererar ström, kollapsar magnetfältet, och detta kollapsande fält inducerar en ström som kommer att flyta i motsatt riktning. Man kan eliminera denna risk genom H-bridge.
34. Beskriv följande kommunikationsinterface RS232, SPI, I2C genom att svara på: synkron/asynkron, antal ledningar och vad de användas till, typ av buss, spänningsnivåer kommunikationshastighet.

RS232	SPI (Serial Peripheral Interface)	I2C (Inter Integrated Circuit)
<ul style="list-style-type: none"> Asynkron (har ingen klocka). 3 och 5 ledningar. 	<ul style="list-style-type: none"> Synkron (innehåller en klocka). 	<ul style="list-style-type: none"> Synkron (innehåller en klocka).

<ul style="list-style-type: none"> • Spänningsnivåer 3 till 15V och -3 till -15V. • 38,4kbits/s 	<ul style="list-style-type: none"> • Gränssnitt med 4 ledningar och en ground (GND). • Serial Data Out (SDO, Master Out Slave In (MOSI)) • Serial Data In (SDI, Master In Slave Out (MISO)) • Select Signal (SS Slave Select) • Enkel gränssnitt, enkelt programmeras av Bit-Banging. • Master mode 40Mbits/s och slave mood 24Mbits/s 	<ul style="list-style-type: none"> • Två trådar gränssnitt och en ground (GND) • Serial klock (SCLK) • Serial Data (SDA) • Master: klocka, starta kommunikation. • Enkel Maskinvara, komplex drivrutin. • 100kbits/s standardmode och 400kbits/s snabbt mode. Långsammare än SPI.
---	--	---

35. En dator med RS232 serieport skall anslutas till mikrokontroller STM32 som matas med 3,3V matningsspänning. Vad är problemet? Hur löser man det?
- +V är vanligtvis 3,3 V och R vanligtvis 100 K Ω . Dessa värden beror på applikationen och den använda hårdvaran. En driver IC behövs för att matcha den högre spänningen av datorn till den lägre spänningen av mikrokontrollen.
36. Förklara vad som menas med hard, firm och soft deadline.
- Hard deadline: om deadline saknas kan den orsaka katastrofala konsekvenser för systemet under kontroll.
 - Firm deadline: om deadline saknas, kan den orsaka några skador på systemet men resultatet har inget värde.
 - Soft deadline: om deadline saknas, kan den orsaka några skador på systemet och utgången har fortfarande ett visst värde även om det orsakar en prestandaförsämring.
37. Ge exempel på en hard, firm respektive soft deadline i ett inbyggt system.
- Hard deadline: flygplanssensorer.
 - Firm deadline: robotmonteringslinjer
 - Soft deadline: programvara för leverans av ljud och video för underhållning.
38. Vad är skillnad mellan preemptive och co-operative multitasking?
- Preemptive multitasking: Operative system styr vilken process som kommer att köras.
 - Co-operative multitasking: Den aktuella tasken avgör om den ska sluta köras. Det vill säga att när en task har börjat, exekverar tills den är klar.
39. En tillståndsmaskin (state machine) kan realiseras med hårdvara eller mjukvara, beskriv vilka typer av kretsar som användas i respektive fall samt beskriv principen för den kan realiseras.
- Hårdvara: olika typer av sensorer som kan avgöra när ett tillstånd ska brytas. Exempelvis paketsortering genom olika sensorer (vikt, färger).

- b. Mjukvara: Real-Time Operating System (RTOS), interrupts, timer kan bestämma när ett tillstånd ska brytas. Ping Pong och RTOS labben är ett bra exempel på en state machine.
40. I en realtidsprocess måste vi läsa av ett antal switcher och vidta någon åtgärd. Det tar 0,1 ms att göra detta. När en switch sluts så är en signal hög minst 10 ms. Visa på en tidsaxel med ett exempel vad som kan vara processens Release time, Initiation time, Completion time, Deadline och Period så att vi säkerställer att vi hinner läsa av switchen. Med vilken Rate exekveras processen?
41. Vad menas med Utilization?
- a. CPU-Utilization anger den del av tiden som processorn spenderar på att utföra uppgifterna för en given uppgiftsuppsättning T med n uppgifter.. Utilization formel av en enkel task ges av $U = C_i/T_i$
42. Vilken uppgift har ett realtidsoperativsystem (RTOS) i ett inbyggt system?
- a. RTOS är en mjukvarukomponent som snabbt växlar mellan olika taskar, vilket ger intrycket att flera program körs samtidigt på en singel process core.
43. Ange några skäl till att använda realtidsoperativsystem.
- a. Abstraherar processorns komplexitet.
b. Ger en solid infrastruktur konstruerad av regler och policyer som ger konsekvens och repeterbarhet.
c. Förenklar utvecklingen och förbättrar utvecklarnas produktivitet genom att använda kärnobjekt på hög nivå för att enkelt hantera komplexa funktioner.
d. Integrerar och hanterar resurser som behövs för kommunikationsstackar och mellanprogram (TCP/IP, USB, SDIO, CAN)
e. Optimerar användningen av systemresurser och förbättrar produktens tillförlitlighet, underhållsbarhet och kavlitet.
44. Hur hanterar realtidsoperativsystem multitasking på en enda processor?
- a. Det sker genom en process som heter context switching, där den sker kontinuerligt. När en context switching inträffar då stoppas den exekverande task och processorregister sparas i minnet, processorregistern för nästa tillgängliga tasken laddas in i CPU och den nya tasken börjar köras. RTOS erbjuder task-to-task meddelandeöverföring, synkronisering av task och allokering av delade resurser till tasken.
45. Vad menas med preemptive multitasking?
- a. Att operativ systemet avbryter en task med lägre prioritet utan samarbete från tasken själv för att utföra en task av högre prioritet.
46. I ett realtidsoperativsystem (RTOS) kan en process befinna sig i ett av tre (eller eventuellt flera) tillstånd. Vilka? Beskriv vad processen gör i respektive tillstånd.



47. Vad menas med Hyperperiod i ett system med periodiska processer?
- Det beskriver den minsta tidsintervall efter vilket schedule upprepas. Det vill säga att den är minsta gemensamma multipeln av perioder av alla tasks i den uppsättningen.
48. Beskriv hur processer som schemaläggs enligt principen round-robin schemaläggs.
- round-robin schemaläggning tilldelar varje uppgift en lika stor mängd av CPU-tiden. I sin enklaste form finns tasks i en cirkulär FIFO-kö och när en task tilldelas CPU-tidlöper ut, läggs tasken till slutet av kön och den nya tasken tas från början av kön. En variant av round-robin-schemaläggning är att tillhandahålla prioritetsbaserad schemaläggning, där task med samma prioritetsnivåer får lika mycket CPU-tid.
49. Vi har följande processer med angivna prioriteter (1 är högst) och exekveringstider. Visa med en tidsaxel över en hyperperiod hur processerna exekveras om vi använder prioritetsdriven schemaläggning.
- Finns i Ipad!!
50. Vi har följande processer med angivna exekveringstider och perioder. Visa med en tidsaxel över en hyperperiod hur processerna exekveras om vi ger processerna prioriteter och schemalägger enligt principen Rate –Monotonic Scheduling (RMS).
- Finns i Ipad!!
51. Vi har följande processer med angivna perioder och exekveringstider. Exekveringen schemaläggs enligt principen Earliest-Deadline first (EDF). Alla processer är redo att starta vid tiden noll. Hur lång hyperperiod har vi i detta exempel?
Visa hur processerna blir schemalagda under de 100 första tidsenheterna.
- EDF är en dynamisk prioritetsschemalägningsalgoritm. Vid körning tilldelas den färdiga uppgiften som har närmast deadline högsta prioritet och körs.
 - Hyperperiod = $\text{lcm}(30, 40, 50) = 600$
 - $U = \text{exekveringstid} / \text{period} = 10/30 + 10/40 + 20/50 \approx 0,983$
deras deadline möts eftersom $U \leq 1$
- | Process | Period | Exekveringstid |
|---------|--------|----------------|
| P1 | 30 | 10 |
| P2 | 40 | 10 |
| P3 | 50 | 20 |
52. Vad menas med critical sections och mutual exclusion i realtidssystem som hanterar flera processor?
- Critical sections hänvisar till en tidsperiod när tasken får åtkomst till en delad resurs, till exempel delat minne
 - Mutual exclusion innebär att endast en process kan vara inne i critical sections när som helst. Om några andra processer kräver den critical delen måste de vänta tills den är ledig.
53. Vad menas med deadlock? Ge exempel på hur det kan inträffa.
- Deadlock är en situation där en uppsättning processer blockeras eftersom varje process håller en resurs och väntar på en annan resurs som förvärvats av någon annan process. Ett exempel uppstår i operativsystem när det finns två eller flera processer som innehåller vissa resurser och väntar på resurser som innehåvs av andra.
54. För att meddela sig mellan processer i ett realtidssystem kan man använda semaforer eller mailboxar. Redogör för skillnaden mellan dessa, samt ge exempel på när det vara lämpligt att använda det ena eller andra systemet.

- a. En semafor är en hink för att lagra en eller flera nycklar. Alla processer som använder semafor måste skaffa en nyckel innan den kan fortsätta att köras. Det finns två typer av semaforer: räknande semaforer och binära semaforer. Semaforer kan användas för att skydda resurser och förhindra att flera tasks kommer åt en resurs samtidigt.
- b. En mailbox är en kommunikationsmekanism som gör att meddelanden kan utbytas mellan processer. Data kan skickas till en mailbox av en process och hämtas av en annan. Konceptuellt beter sig mailboxar som riktiga mailboxar.

55. Beskriv skillnaden mellan processes och threads?

- a. En process är en exekverbar instans av ett program som lagras och arbetar i sitt eget skyddade minnesutrymme. (tilldelningsenhet).
- b. Threads är en ström av utförande som kan dela utrymme med andra. (Utförandeenhet).

56. Förklara vad som menas med att tasks kan vara independent, cooperating och competing.

- a. Independent processor arbetar oberoende av sig själva. De påverkar inte och kan inte påverkas av andra processer som körs i operativsystemet; De delar inte heller data med några processer eller system.
- b. Cooperating processor kan påverkas av andra processer och i sin tur påverka andra processer inom operativsystemet. Dessa processer delar sin data med andra processer och system