DA267A, Programmering av inbyggda system
Malmö Universitet, Fakulteten för Teknik och Samhälle

Tentamen, 2.5 högskolepoäng
December 18, 2023, 14:15 - 18.15

Du ska inte använda någon bok, bilder eller online-resurs. Du kan använda en fickräknare. Du får inte samarbeta med andra studenter via mobiltelefon eller chatt eller att prata. Misstänkta fall av samarbete och plagiering kommer att rapporteras och tentan anses ogiltig.

Du kan svara på svenska eller engelska, men engelska är att föredra. Oroa dig inte om din engelska inte är perfekt, den kommer inte att utvärderas, men svaren måste vara förståeliga.

*Tentamen har 8 frågor med en sammanlagd poäng på 25. Krav för betyg 3 (godkänt) är **15** poäng, krav för betyg 4 är **19** poäng och krav för betyg 5 är **23** poäng.*

Du borde svara med papper och penna. Se till att skriva numret på den fråga du svarar på.

Svar som inte kan tolkas på grund av dålig handskrift eller dålig grammatik ger inga poäng.

För frågor, du kan ringa Dario till 0723756073.

Tips: läs igenom HELA tentamen först och lägg upp en strategi för i vilken ordning ni löser uppgifterna – tex de ni tycker är enkla först och de svåra i mån av tid.

Fråga J1 (2pt)

Vilka steg behöver du genomföra för att lägga in ett element i mitten av en array? Antag att det finns plats att lägga in minst ett element i arrayen och att det redan finns några element i arrayen. Vi vill inte att ni skriver pseudokod, utan det räcker med en konceptuell beskrivning av vad som behöver göras. (2pt)

ANSWER:

1. Ta reda på vilken plats det nya elementet skall stoppas in. Antag att det är index i.
2. Flytta/kopiera alla element som ligger på plats i eller senare. ett steg "bakåt".
3. Lägg in det nya elementet på plats i.

Fråga J2 (3pt)
Beakta följande funktioner som alla beror på n.

- $f(n) = 400n^2 + 3log_2 n$
- $g(n) = 8n^3 + 2n$
- $h(n) = n + 2$
- $i(n) = n^4 + n^2$
- $j(n) = 3n^2 + 5n$

a. Vilken (eller vilka) av funktionerna är $O(n)$? Motivera kortfattat ditt svar. (1pt)

b. Vilken (eller vilka) av funktionerna är $O(n^2)$? Motivera kortfattat ditt svar. (1pt)
c. Vilken (eller vilka) av funktionerna är $O(n^3)$? Motivera kortfattat ditt svar. (1pt)

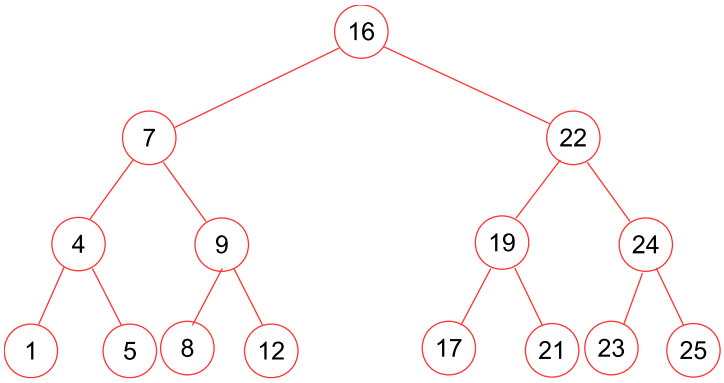
ANSWER:

a. $h(n)$. Övriga funktioner växer fortare än något som är proportionerligt med n.
b. $f(n)$, $h(n)$ och $j(n)$. Övriga funktioner växer fortare än något som är proportionerligt med n^2.
c. $f(n)$, $g(n)$, $h(n)$ och $j(n)$. Funktionen $i(n)$ växer fortare än något som är proportionerligt med n^3.

Fråga J3 (2pt)

Skapa ett balanserat binärt sökträd som innehåller 15 noder med valfria värden.

ANSWER:
Svar: Trädet kan till exempel se ut så här.



Fråga J4 (4pt)

Sortera följande lista med 16 element med hjälp av algoritmen mergesort.

8, 7, 16, 2, 14, 25, 23, 11

Är detta värsta-fallet för mergesort, det vill säga det fall där sortering av 8 element kräver maximalt antal jämförelser? Motivera kortfattat ditt svar.

Om det inte är värsta-fallet, skapa ett värsta-fall med samma värden som i listan ovan.

ANSWER:

8, 7, 16, 2, 14, 25, 23, 11

8, 7 16, 2 14, 25 23, 11

8 7 16 2 14 25 23 11

7, 8 2, 16 14, 25 11, 23

2, 7, 8, 16 11 ,14, 23, 25

2, 7, 8, 11, 14, 16, 23, 25

I plan to give 2p for the merge sort application, 1p for explanation whether this run of merge sort is the worst case behavior of merge sort, and 1p for construction of worst-case scenario.

Detta är inte värstafallet för mergesort då värsta-fallet kräveratt största och näst största värdet tillhör olika listor i varje merge-iteration där listorna har mer än 1 element vardera. . Om man byter plats på 25 och 16 i den ursprungliga listan skulle man ha ett värsta-fall.

8, 7, 25, 2, 14, 16, 23, 11

8, 7 25, 2 14, 16 23, 11

8 7 25 2 14 16 23 11

7, 8 2, 25 14, 16 11, 23

2, 7, 8, 25 11 ,14, 16, 23

2, 7, 8, 11, 14, 16, 23, 25

Fråga J5 (2pt)

Den här frågan testar din kunskap om tillståndsmaskiner.

a. Vilka tre ”komponenter” ingår i en finit tillståndsmaskin (Eng: Finite State Machine)? (1p)
b. Vad är en *Mealy-Moore* maskin? (1pt)

a. Tillstånd, överföringar mellan tillstånd och villkor för överföringar.
b. I en Moore-maskin körs kod i tillstånden medans kod körs övergången mellan tillstånd i en Mealy maskin. En Mealy-Moore-maskin är en kombination av dem båda.

Fråga D1 (3pt)

EN: Describe the way bootloading works on an ESP32. Describe each step including first and second stage bootloaders.

SE: Beskriv hur startladdning fungerar på en ESP32. Beskriv varje steg inklusive bootloaders i första och andra steget.

ANSWER:

First-stage bootloader: configures the access to the external flash memory and, if required, stores on it new data coming from the serial/USB port. Once finished, it accesses the flash memory (at address 0x1000) and loads and executes the second-stage bootloader.

Second-stage bootloader: reads the partition table at address 0x8000 and searches for app partitions. It decides which application must be executed based on the content of the *otadata* partition: if this partition is empty or doesn't exist, the bootloaded executes the application stored in the factory partition.

This allows to implement an over-the-air (OTA) application update process: the application code downloads the new version of your application (e.g. from web) and stores it in a new app partition. Once the upload is completed, the id of the partition is saved in otadata and the chip is rebooted; the bootloader will execute the new version.

Fråga D2 (2pt):

EN:

a) What happens when a function is declared with the keyword "extern" in C? (1pt)
b) What happens when a variable is declared with the keyword "extern" in C? (1pt)

SE:

a) Vad händer när en funktion deklareras med nyckelordet "extern" i C? (1 pkt)
b) Vad händer när en variabel deklareras med nyckelordet "extern" i C? (1 pkt)

ANSWER:

a) Nothing particular. An extern function is a function that is defined somewhere else (in another C file). Anytime a **function** is *declared*, it is **by default extern** (even if one doesn't use the extern keyword). In fact, functions are always global unless declared static.
b) A **variable** declared as *extern* in a module (c-file) means that the variable is **defined** in another module. The memory space is thus reserved in another module, but the compiler knows the type even though it does not reserve any memory space. The variable is implicitly global.

Fråga D3 (5pt):

EN:

You are in charge of the software development of an antitheft device for bikes based on the ESP32. The device senses the presence of the user's phone using Bluetooth, when it cannot detect it, it assumes that the user is away. The device has also an accelerometer. When enough movement is detected, and the user's phone is not in range, it will assume that someone is trying to steal the bike and will trigger an alarm.

Your new intern from Malmö University (here in the picture) has laid out the architecture of the software that is going to run on the microcontroller. He's a bit unexperienced and the architecture is messy (see figure A).

A) Organize the architecture in layers. Use figure A below as an input. (2pt)

The product is released, but after a few days you start receiving complaints from customers because the device sometimes stops working when it should start the alarm. The hardware team blames the software team. They say that you have too many FreeRTOS tasks running concurrently and the CPU cannot handle the load. You review the periodic tasks your team has implemented:

Task1: Bluetooth stack
lines of code: 500 kloc (thousands of lines of code)
period: 100 ms

Task2: acceleration sampling
lines of code: 10 kloc
period: 10 ms

Task3: movement detection
lines of code: 20 kloc
period: 500 ms

Task4: phone detection
lines of code: 8 kloc
period: 250 ms

Task5: control
lines of code: 32 kloc
period: 100 ms

Task6: alarm
lines of code: 1 kloc
period: 10 ms

The scheduling strategy is Rate-monotonic Scheduling, the CPU runs at 160MHz. All tasks run on the same core.

B) Show the hardware team that it's not your fault! Prove that the tasks can be scheduled. (3pt). Hint: $6 \cdot \left(2^{\frac{1}{6}} - 1\right) = 0.73$

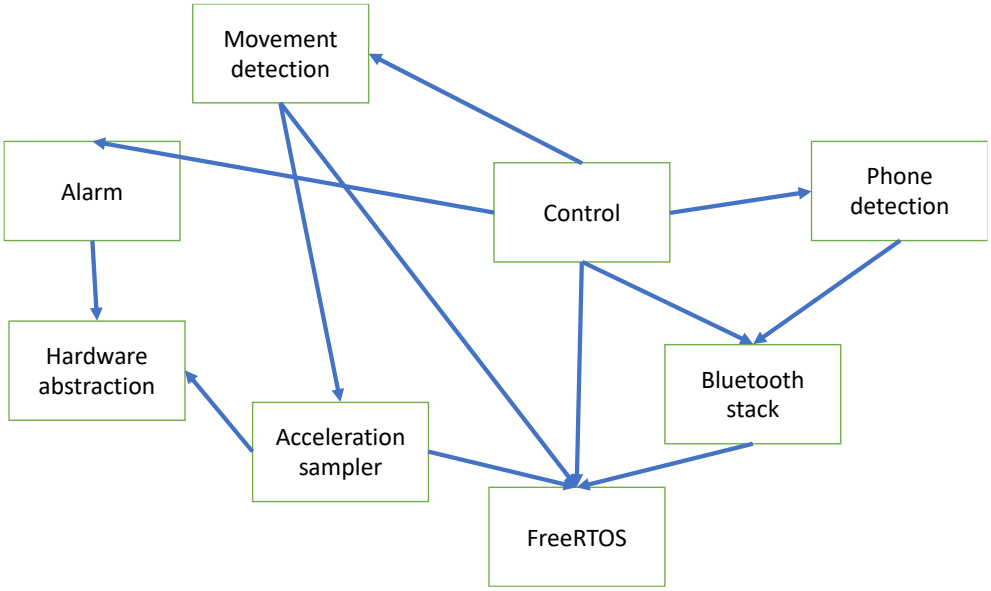


Figure A: software architecture.

SE:

Du är ansvarig för mjukvaruutvecklingen av en stöldskyddsenhet för cyklar baserad på ESP32. Enheten känner av närvaron av användarens telefon med hjälp av Bluetooth, när den inte kan upptäcka den antar den att användaren är borta. Enheten har också en accelerometer. När tillräckligt med rörelse upptäcks och användarens telefon inte är inom räckhåll, kommer den att anta att någon försöker stjäla cykeln och utlöser ett larm.

Din nya praktikant från Malmö Universitet (på bilden) har lagt upp arkitekturen för programvaran som ska köras på mikrokontrollern. Han är lite oerfaren och arkitekturen är rörig (se figur A).

A) Organisera arkitekturen i lager. Använd figur A nedan som indata. (2 pkt)

Produkten släpps men efter några dagar börjar du få klagomål från kunder eftersom enheten ibland slutar fungera när den ska starta larmet. Hårdvaruteamet skyller på mjukvaruteamet. De säger att du har för många FreeRTOS-uppgifter som körs samtidigt och CPU:n kan inte hantera belastningen. Du granskar de periodiska uppgifter som ditt team har implementerat:

Uppgift 1: Bluetooth-stack
kodrader: 500 kloc (tusentals rader kod)
period: 100 ms

Uppgift 2: accelerationssampling
kodrader: 10 kloc
period: 10 ms

Uppgift 3: rörelsedetektering
kodrader: 20 kloc
period: 500 ms

Uppgift 4: telefondetektering
kodrader: 8 kloc
period: 250 ms

Uppgift 5: kontroll
kodrader: 32 kloc
period: 100 ms

Uppgift 6: larm
kodrader: 1 kloc
period: 10 ms

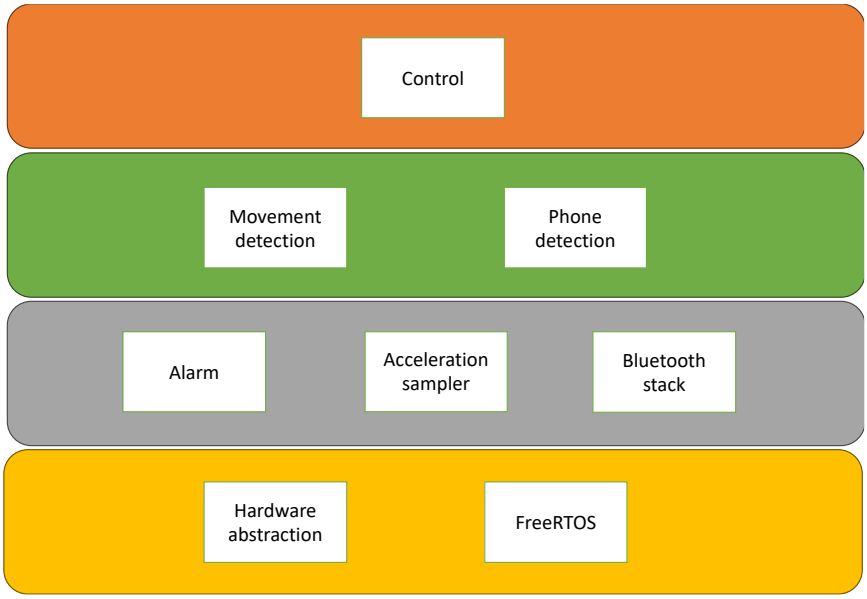
Schemaläggningsstrategin är Rate-monotonic Scheduling, CPU:n körs på 160MHz. Alla uppgifter körs på samma kärna.

Visa hårdvaruteamet att det inte är ditt fel! Bevisa att uppgifterna kan schemaläggas. (3 pkt). Tips: :
$6 \cdot \left(2^{\frac{1}{6}} - 1\right) = 0.73$

ANSWERS:

A)

Several answers are possible, but they all must respect dependency constraints, with lower layers having modules that are used by other modules. An example:



B)

If RMS is used, according to the "utilization bound test" ($U <= n*(2^{1/n}-1)$ where n=5), the utilization factor should not be more than 0.73. We compute the utilization factor for those 5 tasks with clock at 160MHz and assuming that each line of code translates into machine 10 instructions, with each instruction using only 1 clock cycle.

Task 1: u= 0,3125, Task 2: u= 0,0625, Task 3: u= 0,0025, Task 4: u= 0,002, Task 5: u= 0,02, Task 6: u= 0,00625.

Total U = 0,40575, which is less than 0.73. Take this hardware team!

Fråga D4 (2pt):

EN: Show how UML diagrams can be used for the 4+1 view architectural model. Fill in the following table by placing the UML diagrams from the list below into the right column. For each correct answer you get +0.3pt. For each wrong answer, you get -0.3pt.

SE: Visa hur UML-diagram kan användas för arkitekturmodellen med 4+1-vy. Fyll i följande tabell genom att placera UML-diagrammen från listan (se ner) i den rätta kolumnen. För varje rätt svar får du +0,3 pkt. För varje fel svar får du -0,3 pkt.

View	Logical view	Process view	Development view	Physical view	Scenarios

UML – Diagram					

UML diagrams:

Class Diagram, Object Diagram, Activity Diagram, State Diagram, Timing Diagram, Sequence Diagram, Component diagram, Package diagram, Deployment diagram, Use case diagram.

ANSWER:

View	Logical view	Process view	Development view	Physical view	Scenarios
UML – Diagram	Class, Object Diagram	Activity, State, Timing, Sequence Diagram	Component, Package diagram	Deployment diagram	Use case diagram