

Övning 6. Bibliotek

Bakgrund

Resistorer är en elektronikkomponent som kan användas i elektriska kretsar. Kort beskrivet kan man säga att den är ett motstånd mot ström i en elledning. Den kan jämföras med en kran i ett vattenrör. Om man öppnar kranen lite är motståndet mot vattnets rörelse (ström) stort och det rinner lite vatten i röret. Om man skruvar upp kranen mer och mer minskar motståndet och vattnets rörelse (strömmen) ökar. Man kan koppla resistanser efter varandra i en serie och då kallas kopplingen seriell koppling. Om man kopplar motståndet "bredvid" varandra kallas det parallellkoppling. Dessa kopplingar med flera resistorer kan alltid ersättas av en resistor med samma resistans som de ursprungliga och den "ersättningsresistansen" kan beräknas enligt olika formler. Det är att jämföra med att om man har flera halvöppna kranar efter varandra i en vattenledning kan alla kranarna ersättas en kran med samma motstånd som summan av de ursprungliga. Se tex denna webbsida för lite mer information om det hela:

<http://www.ict.kth.se/courses/IF1330/online/circuit/index.htm>

När man i praktiken ska bygga elektronik och använda resistorer finns dessa inte att köpa i värden i en kontinuerlig lista. De indelas i ett bestämt antal värden och dessa listor med möjliga värden är standardiserade i sk E-listor. Vi ska här använda de resistansvärden som återfinns i serien E12.

Uppgift

Denna övning ska ni lösa i grupper om 3 och 3 studenter. Ni ska skriva 3 dynamiska bibliotek, ett vardera, med funktioner enligt nedan. Dessa ska sedan länkas ihop med ett huvudprogram där ni testar alla bibliotek med lämpliga anrop till funktioner. Det är inte viktigt hur huvudprogrammet beter sig eller hur det är skrivet. Men funktioner från alla bibliotek ska användas och länkas in. Interfacen för funktionerna måste följa beskrivningarna nedan exakt med rätt namn, argument och returvärde.

Efter att man blivit godkänd på de 2 första övningarna på kursen OCH skickat ett mail enligt [dessa anvisningar](#) blir man indelad i en grupp och får veta vilka man ska lösa denna övning tillsammans med. Har du ingen grupp och har fått godkänt på övn 2 samt skickat mailet enligt ovan, hör av dig snarast till [Björne](#).

Har man kommit in i kursens 3-4:e månad (maj för vårtermin och dec för hösttermin) kan det dock bli svårt att gruppera in i grupper och det kan hända att man måste vänta till start av nästkommande termin för att bli indelad i grupp.

OBS! Uppgiften ska lösas i samarbete och i grupp för att man ska bli godkänd! Löser man uppgiften på egen hand kommer man inte att bli godkänd på detta moment. Uppstår det problem i gruppen hör ni av er till [Björne](#).

Bibliotek 1, libresistance.so

Skriv ett bibliotek, libresistance.so, med funktioner för att beräkna den totala resistansen för olika kopplade resistanser. Resistanserna ska kunna kopplas parallellt eller i serie. Man behöver inte räkna på blandade kopplingar.

float calc_resistance(int count, char conn, float *array);

Där:

- count: Antal komponenter
- conn: Seriellt eller parallellt kopplade komponenter [P | S]
- *array: En pekare på en array av komponentvärden som är lika stor som count.
- Värdet 0 skall returneras om något motstånd är noll vid parallellkoppling, dvs $R1 || R2 = 0$, om $R1$ eller $R2$ är 0 Ohm.
- Biblioteket får inte krascha om en "nollpekare" skickas till funktionen, dvs om array=0.
- Om argumenten är felaktiga skall funktionen returnera -1.

Returvärdet är den resulterande resistansen

Bibliotek 2, libpower.so

Ström som passerar ett motstånd värmer upp motståndet med en viss effekt (P). Effekten kan beräknas med hjälp av spänningen och ström eller spänning och motståndsvärdet enligt dessa formler:

$P = U * I$ (Spänning gånger strömmen)

$P = U^2 / R$ (Spänning i kvadrat delat i resistansen)

Skriv ett bibliotek, libpower.so, med funktioner för att beräkna den totala effektutvecklingen i en krets med en spänningskälla kopplad i serie med en resistans:

float calc_power_r(float volt, float resistance);

float calc_power_i(float volt, float current);

Bibliotek 3, libcomponent.so

Skriv ett bibliotek, libcomponent.so, med funktionen:

```
int count = e_resistance(float orig_resistance, float  
*res_array );
```

En funktion som beräknar vilka tre seriekopplade resistorer i E12-serien som närmast ersätter den resistans som skickas med.

- orig_resistance är ersättningsresistansen.
- *res_array är en pekare till en array med 3 resistorer som ska fyllas med värden ur E12-serien.
- count är hur många resistorer ur E12-serien som behövdes för att ersätta orig_resistance Om inte alla 3 komponenterna behövs ska de som inte används fyllas med värdet 0. count kan anta värde mellan 0 och 3

Huvudprogram

Bygg ovanstående bibliotek och skriv ett huvudprogram, **electrotest**, som använder biblioteksfunktionerna. Programmet kan tex fråga efter spänningskälla, kopplinssätt, antal komponenter samt värdet på varje komponent och returnera resulterande resistans, effektutveckling samt ersättningsresistanser från E12-serien.

%./electrotest

Ange spänningskälla i V: 50

Ange koppling[S | P]: S

Antal komponenter: 3

Komponent 1 i ohm: 300

Komponent 2 i ohm: 500

Komponent 3 i ohm: 598

Ersättningsresistans: 1398 ohm

Effekt: 1.78 W

Ersättningsresistanser i E12-serien kopplade i serie: 1200, 180, 18

Funktionerna får inte använda statiska arrayer (float values[2000]) för att hålla reda på de angivna komponenternas värden. Använd dynamisk minnesallokering med någon variant av malloc och free. I övrigt går det bra med statiskt allokerade tabeller (tex för att hålla E12-värdena).

Redovisning

Alla studenter lämnar in en individuell rapport med dessa 3 delar:

Del 1

Implementerad det bibliotek du fått enligt ovan samt ett fullständigt program där du länkar in ditt bibliotek så man kan se att det fungerar. Vad programmet gör är upp till dig men det ska anropa samtliga funktioner i ditt bibliotek och det ska klart framgå att biblioteket fungerar enligt beskrivningen.

Redovisa en beskrivning av det egna biblioteket och applikationen vad gäller:

- Funktion och användning
- En algoritmbeskrivning
- Hur du kompilerat och testat det i ett eget program. Beskriv vilka kommandon du använder för att kompilera biblioteket och hur du sedan använder biblioteket i ditt huvudprogram.
- Beskriv de växlar du använt till kommandona ovan och vad var och en av dessa har för funktion

Del 2

Skriv en Makefile med dessa regler:

- lib, för att bygga enbart biblioteket
- all, För att bygga både programmet och biblioteken där biblioteken läggs i en egen katalog, lib, under den man är just nu, tex /home/bl/electro/lib/. Här ska programmet länkas för att använda de lokala biblioteken.
- install. Här kopierar du både programmet och biblioteken till lämpliga kataloger (tex /usr/bin/ och /usr/lib/) och länkar så att programmet använder de publika biblioteken.

Redovisa en beskrivning av hur du länkat in alla 3 bibliotek och använt det i ett huvudprogram enligt ovan. Beskriv också vilka du jobbat med.

Beskriv vilka kommandon och växlar du använder för att kompilera biblioteken och hur du sedan använder biblioteken i ditt huvudprogram.

Alla 3 studenter skriver var sitt bibliotek. Ni får gärna hjälpa varandra men inte göra jobbet åt varandra. -Fråga och hjälp varandra. Sedan hjälps ni åt att skriva ett gemensamt huvudprogram och Makefile för att kompilera och använda alla biblioteken.

Bifoga ett arkiv med källkod och Makefile för att kunna kompilera projektet till en körbar fil.

Del 3

Skriv en kort beskrivning av hur samarbetet gått samt en reflektion över vad som krävs för att ett sådant här arbete ska fungera även i större skala med betydligt större skara utvecklare och större kodbas.

Del 2 i rapporten får gärna vara identiska.