

Projekt av Johan Svensson

Kontaktinformation: Johan Svensson <johans7@kth.se>

Under min arbetskarriär, studier och fritid har jag producerat skript och programvara för att utföra olika uppgifter i olika programmeringsspråk och datormiljöer. Dessa beskrivs i en lista nedan, länk till projekt hittas i länkförteckningen.

Projektinnehåll

Stödverktyg för ProBIND	1
Examensarbete, KTH	1
Skoluppgift i kursen inbyggda system	3
Skoluppgifter i mobila applikationer och trådlösa nät.....	3
Skoluppgift i VHDL-design	3
Skoluppgift i objektorienterad programmering	4
Uppgradering av högtalarelement i befintlig låda.	4
Projektkurs på KTH, andra läsåret.....	4
IOT Sync.....	5
EUMT – Easier Unix Management Tool.....	5
Netgear router, no internet – reboot.....	5
Projektkurs på KTH, första läsåret.....	6
Programkod för en projektvideo	6
PIC18F1823 Projekt blomvattnare	6
Tidigare arbetsuppgifter.....	6
TeleComputing - Unix administratör.....	6
Bwin – Linux administratör.....	7
Länkförteckning.....	8

Stödverktyg för ProBIND

Använt programspråk: Bash, PHP.

Länk till projekt: Programkod [1].

Som ett stödverktyg till ProBIND för uppställning av domäner (DNS – Domain Name System) och migrering av domäninformation från en databas till en annan, vilket är en manuell åtgärd med flera manuella steg realiserades skript för automatisering vilket blev tre program [1]. Det första skriptet inaktiverade domäner som ej var delegerade från toppdomänen, det innebär att DNS förfrågningar om domäner ej når den avsedda namnservern utan det var delegerat till en annan domänservr på internet, det andra skriptet raderade domänerna från databasen, det tredje skriptet flyttade domänerna med samtliga poster till en ny databas. Två skript programmerades i Bash samt ett realiserades i PHP.

Examensarbete, KTH

Använt programspråk: C++ (Visual studio 2012).

Länk till projekt: Rapport [2].

I examensarbetet utvecklades en prototyp för styrning av två stegmotorer med ModbusTCP protokollet samt bildanalys från en ljussensor med OpenCV biblioteket. Då drivrutiner saknades för ModbusTCP protokollet utvecklades en egen drivrutin för kommunikation mellan datorn och stegmotorn där Modbus protokollet kapslades in i ethernet-ramens applikationsdata. Programalgoritmen delades upp i fyra trådar, två stycken motor-, en kamera- och en huvudtråd. Trådarna hade olika exekveringstider och periodtider. För kommunikation mellan trådar användes delade meddelanden (mutex objekt) för att garantera exklusiv åtkomst till objektet. Motortrådets uppgift var att kommunicera med stegmotorn via ModbusTCP, kameratrådens uppgift var att inhämta information från ljussensorn och huvudtrådens uppgift var att analysera den inkomna informationen och styra motor- och kameratråden med hjälp av meddelanden.

För låsning av ett meddelandeobjekt användes en privat metod getLock, vilket ses i figur 1.

```
bool Pkg_thread::getLock(){
    int retrycount = 1000;
    bool mutex_lock = false;
    //try to lock the mutex
    for(int i = 0; i < retrycount;i++){
        if(mu.try_lock()){
            mutex_lock = true;
            break;
        }
    }
    //check if lock was successful.
    if(mutex_lock == false){
        printf("MUTEX NOT LOCKED\n");
        return false;
    }
    return true;
}
```

Figur 1. Metoden getLock.

En kommunikationstråd mellan motorn i x-led och huvudtråden visas i figur 2.

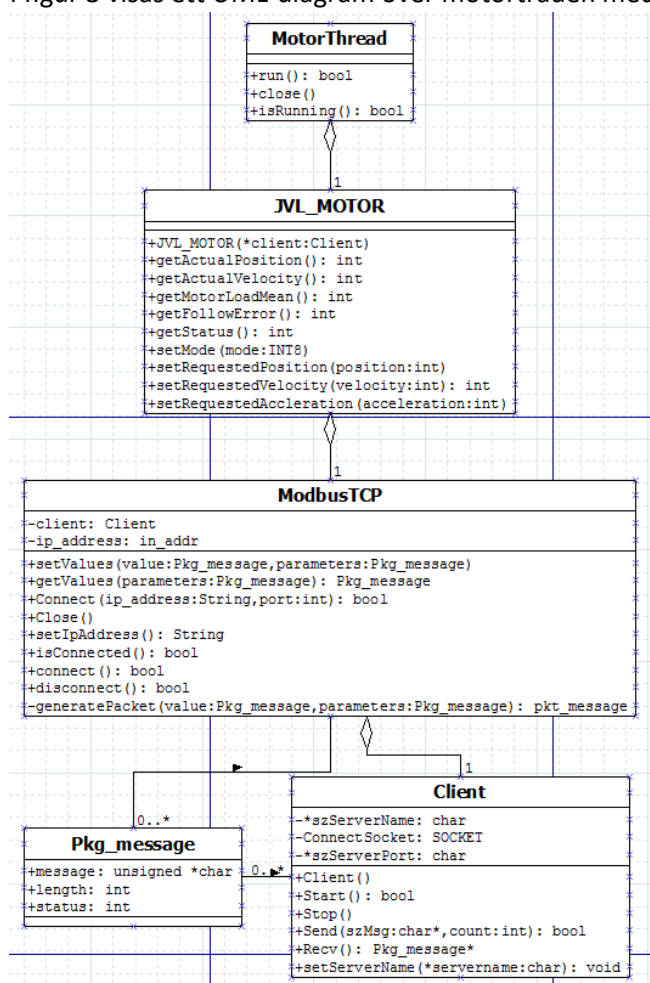
```

_MotorX: NEW MESSAGE FROM MAINTHREAD
_MotorX: message is: SET POSITION 19170
_MAINTHREAD: NEW MESSAGE FROM MOTOR_X
_MAINTHREAD: message is: SET POSITION OK
_MotorX: NEW MESSAGE FROM MAINTHREAD
_MotorX: message is: GET POSITION
_MAINTHREAD: NEW MESSAGE FROM MOTOR_X
_MAINTHREAD: message is: GET POSITION 22554
_MotorX: NEW MESSAGE FROM MAINTHREAD
_MotorX: message is: GET POSITION
_MAINTHREAD: NEW MESSAGE FROM MOTOR_X
_MAINTHREAD: message is: GET POSITION 19170
_MAINTHREAD: MOTOR IS IN POSITION!!

```

Figur 2. Kommunikation mellan huvud- och motortråden.

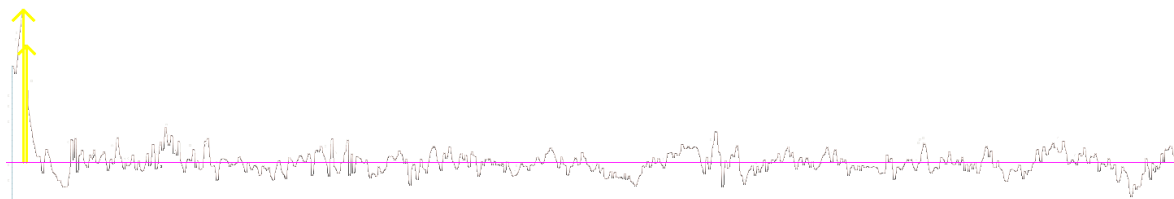
I figur 3 visas ett UML-diagram över motortråden med tillhörande klasser.



Figur 3. Klassschema av motortråden.

Ett fotografi inlästes med OpenCV biblioteket och behandlades med Canny's kantdetekteringsalgoritm, därefter togs de tjugo högsta punkterna ut, om tre närliggande pixlar hade ungefärlig y-koordinat registrerades det som en vektor och en gul pil placerades på fotografiet. I figur

4 visar ett exempel där metoden applicerats på ett fotografi.



Figur 4. Test med referenslinje och vektordetektion.

Skoluppgift i kursen inbyggda system

Använt programspråk: C.

Länk till projekt: Programkod [3].

I skoluppgiften inbyggda system [4] på KTH ingick fem laborationer där testdriven utveckling användes, målet var att realisera en "turtle" robot [5] som ritade ut linjer efter operatörens kommandon, kommunikation med roboten gjordes med en radiomodul ansluten till en dator. I kursen ingick att kontrollera två stegmotorer med hjälp av en stegmotorkrets, en servomotor för kontroll av en pennas position i höjdled och en radiomodul ansluten till UART där mottagna samt skickade meddelanden gjordes med avbrott. Även RTOS var med som ett moment vid ett laborationstillfälle.

Skoluppgifter i mobila applikationer och trådlösa nät

Använt programspråk: Java (Android studio).

Länk till projekt: Android pulsmätare [6], Java TCP-server [7], Android valutaväxlare [8].

I kursen mobila applikationer och trådlösa nät [9] på KTH utvecklades två applikationer för operativsystemet Android.

Det första programmet som utvecklades var en valutaväxlare mellan två olika valutor där information inhämtades från ECB i XML-format. I rullgardinsmenyn inlästes även respektive lands flagga. Ett användarvänligt GUI och möjlighet att schemalägga nedladdning av valutakursen fanns.

Det andra programmet som utvecklades var en pulsmätarapplikation där skolan hade en pulsmätare som kommunicerade via Bluetooth. Information om pulsen insamlades och när användaren avbröt insamlingen i Android applikationen fanns möjlighet att skicka informationen vidare till en dator. För möjliggörande av kommunikation realiserades en TCP-server i Java som lyssnade på en port och sparade ned inkommande meddelande till fil.

Skoluppgift i VHDL-design

Använt programspråk: VHDL.

Länk till projekt: Programkod [10].

I kursen VHDL-design [11] på KTH ingick fem laborationer där varje laboration var ett bidrag för att slutligen syntetisera en CPU (Central Processing Unit). Utvecklingskittet var från leverantören Altera där applikationerna Quartus och Modelsim användes. En ALU (Aritmetisk logisk enhet), kontroller, databuffer, multiplexer, ROM, RAM och registerfil realiserades och syntetiserades med tillhörande testbänkar.

Skoluppgift i objektorienterad programmering

Använt programspråk: Java (IntelliJ).

Länk till projekt: Programkod [12].

I kursen objektorienterad programmering [13] studerades metoder för programutveckling enligt ett objektorienterat programmmönster. Där beskrevs flöden med klassdiagram enligt UML, relationer, arv, undantag, polymorf, strömmar och händelsestyrda applikationer. Som sista laboration i kursen utvecklades pokerspelet Texas hold em poker. Programmet utvecklades enligt Model-View-Controller där jag utvecklade samtliga delar förutom den grafiska layouten..

Uppgradering av högtalarelement i befintlig låda.

Använt program: Orcad PSpice.

Länk till projekt: Hemsida [14].

Som ett sommarprojekt byggde jag ihop en ny högtalarlåda med tillhörande element. Elementen ritades upp i programmet Orcad Pspice där baselementet och diskantens småsignalschema analyserades. Filtret som skickade låga frekvenser till baselementet och höga frekvenser till diskanten analyserades. I figur 5 gjordes en "AC-sweep" för att se filtrets karaktäristik vid olika frekvenser, där blå linje var insignalens amplitud, blå linje var lågpas-filtret och röd linje var högpass-filtret.



Figur 5. AC-sweep av delningsfilter.

Projektkurs på KTH, andra läsåret

Använt programspråk: PHP, Java (Android studio)

Andra teknologier: MySQL, FreeBSD, Nginx.

Länk till projekt: Rapport [15], video [16], hemsida [17].

I kursen projektkurs inom elektroteknik [18] realiserades en EKG-läsare med stöd för Internet of Things (IoT). Jag var projektledare för gruppen och hade övergripande ansvar för olika delarna inom projektet. En analog krets hade tre insignaler som anslöts till elektroder, en anslöts i höger fot, den andra i vänster- och den tredje i högerarm. Insignalen förstärktes ungefär 600 gånger för att avläsas av en mikrokontroller (MCU), spänningsområdet för avläsning var mellan 0-3 Volt. Efter att den

analoga signalen konverterats till digitala värden skickades de vidare till en mobiltelefon via Bluetooth där de presenterades i en graf på skärmen. Sedan skickades informationen via internet till en server för lagring i en databas. Servern publicerade även informationen på en hemsida där den presenterades som en graf.

IOT Sync

Använt programspråk: Bash, PHP.

Andra teknologier: MySQL, FreeBSD, Linux.

Länk till projekt: Programkod [19].

Som ett projekt utvecklade jag en lösning över en helg för synkronisering av enheter för IoT, det var tänkt att använda produkten tillsammans med EKG-läsaren vilket uteblev, istället blev det ett fristående projekt. Tanken var att en slav "ringer" hem till sin master för delgivning av information. Både slaven och mastern har inkommande och utgående kataloger. När mastern vill inhämta information placeras en fil med beskrivning av vilka uppgifter som ska inhämtas. När den schemalagda uppgiften exekveras på slaven skickas och hämtas information till sin master, finns inkommande meddelanden avhandlas dessa och ett svar läggs i utkorgen. Vid nästa schemalagda uppgift skickas meddelandet tillbaka till mastern som sedan avläser meddelandet. Tanken var möjliggöra så att EKG-läsare med IoT stöd skulle kunna knytas ihop till en lösning, eller alternativt att andra IoT enheter kunde utnyttja synkroniseringsprogramvaran.

EUMT – Easier Unix Management Tool

Använt programspråk: C.

Andra teknologier: Linux.

Länk till projekt: Programkod [20].

Efter avslutad termin påbörjade jag ett projekt för att fördjupa kunskaperna inom programmeringsspråket C. En vän till mig hade svårigheter med administration av sin Linux dator varpå utveckling av ett manageringsverktyg påbörjades. Programmet var tänkt att ersätta inloggningsskalet i terminalen, det innebar att vid inloggning med SSH (Secure Shell) visades alternativ som användaren hade möjlighet att välja av. Programmet läste in en konfigurationsfil där samtliga val med parametrar var definierad, dynamisk allokering av minnesutrymme användes. Det kan för övrigt användas inom organisationer där begränsad erfarenhet av Linux finns, där administratörer har möjlighet att göra rätt uppgift beroende på alternativ som finns definierad i konfigurationsfilen.

Netgear router, no internet – reboot

Använt programspråk: Bash.

Andra teknologier: Linux.

Länk till projekt: Programkod [21]

Ett skript utvecklades för testning av internet åtkomsten då Netgear routern ibland tappade kontakten och en omstart behövde göras. Skriptet exekveras i en Linux-miljö som exempelvis ett schemalagt jobb, det första skriptet gör var att testa om google svarar på en Ping förfrågan, om ej svar fås skickas ett kommando till Netgear routern för omstart.

Projektkurs på KTH, första läsåret

Använt programspråk: C (Arduino), Bash, PHP.

Andra teknologier: MySQL, FreeBSD, Apache.

Länk till projekt: Video [22]

I kursen projektkurs inom elektroteknik och internetteknik [23] utvecklades en prototyprobot som åkte runt i ett rum och kartlade väggar och objekt däri. Jag var projektledare och hade övergripande ansvar för projektet. Robotens centrala styrenhet var en Arduino som styrde en motorstyrkrets som i sin tur strömförde stegmotor som var anslutna till larvfötter. Tre ultraljudssensorer var anslutna till Arduino, en till vänster, en till höger och en rakt fram. Med hjälp av ultraljudssensorerna navigerade roboten runt i rummet och sparade objekt och rummets väggar, därefter hämtades de in i en dator och sedan laddades kartinformationen upp på en hemsida som sedan ritade ut rummet och presenterade det i en figur.

Programkod för en projektvideo

Använt programspråk: C.

Andra teknologier: Linux.

Länk till projekt: Programkod [24].

I samband med projektkursen under första året producerades en kortfilm om projektet och som effekt utvecklades en programkod som skrev ut en bokstav i taget på skärmen med en fördefinierad fördröjningsintervall.

PIC18F1823 Projekt blomvattnare

Använt programspråk: C.

Länk till projekt: Hemsida [25]

Efter avslutad kurs i mikrodatorteknik [26] utvecklade jag en blomvattnarprodukt med en PIC processor. Kretsen innehöll en knapp och nio lysdioder. När produkten startas för första gången väljs intervall med knappen, en knapptryckning ger en intervall på ett dygn och en lysdiod lyser upp, två knapptryck ger ett intervall på två dygn och två lysdioder lyser upp, det maximala intervallet var sju dygn, varav totalt sju lysdioder användes för indikering av intervall. När åtta timmar kvarstår börjar en gul lysdiod att blinka och när tiden har passerat lyser en röd lysdiod. Programmet använde avbrott och djup-sov-funktioner.

Tidigare arbetsuppgifter

Här beskrivs kortfattat skript och programvaror som realiserats som ingick i mina arbetsuppgifter.

TeleComputing - Unix administratör

Använt programspråk: Bash.

Andra teknologier: Unix och Linux.

För automatisering av manuella arbetsuppgifter togs olika skript fram, dels schemalagda men även användarinitierade. En kund på TeleComputing hade ett inventarieprogram där innehållet presenterades på en hemsida. Däribland fanns inventarielistor på servrar, nätverksadresser och lösenord för samtliga konton. För uppsökande av nätverksadresser och användarkonton för en server krävdes att en sökning först utfördes, därefter blev servernamnet klickbart varpå man klickade på länken. På informationssidan för servern presenterades nätverksadresserna knutna till servern samt användarkonton, för att inhämta lösenordet för en användare klickade man på en länk varpå en nytt

fönster öppnades med lösenordet. Skriptet som utvecklades loggade in på hemsidan, sökte upp servern i fråga och hämtade ut ipadresser och användarkonton med tillhörande lösenord som tillhörde servern, detta presenterades sedan överskådligt i kommandotolken.

Bwin – Linux administratör

Använt programspråk: PHP, Bash.

Andra teknologier: Linux, Windows.

En programvara som utvecklades var till personalen som jobbade med jour och larmhantering. En hemsida samt ett bakgrundsjobb utvecklades. Varje minut exekverades ett schemalagt jobb som loggade in på en Windowsutdelad mapp, om en fil fanns på den utdelade mappen hade ett larm inkommit, innehållet i filen lästes in och sparades i en databas för att sedan raderas. På hemsidan hade ett nytt larm presenterats, efter att incidenten avhandlats skrevs en journalanteckning om åtgärd. Trettio minuter efter avslutat skift sammanställde det schemalagda jobbet samtliga larm och anteckningar och skickade ett mail till berörda personer om vad som inträffat under jourtillfället. Då juren var bemannad alla timmar på dygnet. Ett skift avslutades klockan 06:00 och för att undvika att väcka jourhavande person för att göra ett mailutskick gjorde programvaran det per automatik.

Andra programvaror och skript realiserades i samband med tjänsten på Bwin, däribland administrationsverktyget som var utvecklat i Bash. Inför ett servicefönster där uppgradering av flera servrar med tillhörande tjänster uppgraderades gjordes anrop mot övervakningstjänsten för att avaktivera larm för servern, och efter att servern med tjänster var uppgraderade och startade aktiverades larmen igen.

Länkförteckning

- [1] J. Svensson, ProBIND migration, https://github.com/payano/ProBIND_migration
- [2] J. Svensson, Mätning av skärgradshöjd på stål, <http://exclude.se/KTH/examensarbete.pdf>
- [3] J. Svensson, STM32 Turtle, https://github.com/payano/stm32_turtle
- [4] KTH, Inbyggda system, <https://www.kth.se/social/course/IS1300/>
- [5] B. Molin, Turtle Art, <https://www.youtube.com/watch?v=ddV22CXOejE>
- [6] J. Svensson, Android Bluetooth Pulse Meter, <https://github.com/payano/BluetoothPulseMeter>
- [7] J. Svensson, Java TCP-server, <https://github.com/payano/Java-TCPServer>
- [8] J. Svensson, Android currency, https://github.com/payano/android_currency
- [9] KTH, Mobila applikationer och trådlösa nät, <https://www.kth.se/social/course/HI1033/>
- [10] J. Svensson, VHDL CPU, https://github.com/payano/VHDL_CPU
- [11] KTH, VHDL-design, <https://www.kth.se/social/course/IL1331/>
- [12] J. Svensson och A. Bodin, Texas Hold Em, https://github.com/payano/texas_hold_em
- [13] KTH, Objektorienterad programmering, <https://www.kth.se/social/course/HI1027/>
- [14] J. Svensson, Uppgradering av högtalarelement i befintlig låda, <http://www.exclude.se/?p=125>
- [15] J. Svensson, B. Markovic och A. Bodin, Projektarbete årskurs 2, http://exclude.se/KTH/rapport_grupp21.pdf
- [16] J. Svensson, A. Bodin och B. Markovic, Video Smart ECG-Reader, University project, <https://www.youtube.com/watch?v=Qmb5UqK7q3o>
- [17] J. M. B. Svensson och A. Bodin, Presentation av EKG-graf, <http://fbsd.exclude.se/>
- [18] KTH, Projektkurs inom elektroteknik, <https://www.kth.se/social/course/HE1035/>
- [19] J. Svensson, IOT Sync, https://github.com/payano/loT_sync
- [20] J. Svensson, Easier Unix Management Tool, <https://github.com/payano/EUMT>
- [21] J. Svensson, Netgear router, no internet – reboot, <http://www.exclude.se/?p=64>
- [22] J. Svensson, F. Atroshi, J. Söderström, Emil och Gustav, Rover 5 room orientation, <https://www.youtube.com/watch?v=KCigeelBw98&feature=youtu.be>
- [23] KTH, Projektkurs inom elektroteknik och internetteknik, <https://www.kth.se/social/course/HE1029/>
- [24] J. Svensson, Code for the project presentation, <http://www.exclude.se/?p=40>
- [25] J. Svensson, PIC18F1823 Programkod (skapad i mplabx), <http://www.exclude.se/?p=10>
- [26] KTH, Mikrodatorteknik, <https://www.kth.se/social/course/HE1028/>