

Labb Inet: Internet/sockets

I denna laboration ska du rätta och förbättra ett bankomatprogram. För att bli godkänd på labben måste ditt program fungera, uppfylla en del krav och vara dokumenterat.

Syftet med labben är att studera internet-orienterad programmering och då speciellt kommunikation via sockets (det finns en [officiell tutorial](#) om sockets att läsa för den som är intresserad). Det är särskilt viktigt att du kan beskriva hur kommunikationen fungerar och därför är det viktigt att ni kan specificera i detalj det så kallade protokollet som de kommunicerande programmen använder.

Exempelkoden är skrivet i java. Det brukar gå att skriva i ett annat språk som t.ex. Python men skriv till föreläsaren (alba) innan du börjar. Särskilt lärorikt kan vara att skriva i t.ex. Go eller Erlang som är konstruerat för att hantera en parallell programmeringsparadigm. Erlang är däremot inte så bra på att hantera strängar och man kan behöva skriva en liten front-end i java för användargränssnittet.

(Denna labb rättas inte på Kattis.)

Bakgrund

Ladda hem [exempelkoden](#). Det finns en serverdel med två klasser `ATMServer`, `ATMServerThread` som datorkonsultföretaget Viebrapadata har skrivit. Klientdelen `ATMClient` har företaget Snilledata skrivit. Programmen fungerar nästan.

Sätt dig in i koden. Kompilera den. Starta en server med `java ATMServer &`. Öppna en ny terminal på samma maskin och starta klienten med `java ATMClient 127.0.0.1`. Prova att köra programmet några gånger. Prova med två samtidigt uppkopplade klienter. Lägg märke till att saldo återställs för den andra klienten. Det tycker inte banken är bra.

Uppgift

De båda företagen är sura på varandra för att det går dåligt att samarbeta. När Viebrapadata släppte en ny version av server slutade klienten att fungera för att servern hade ändrat sin meny så att den skrev ut alternativen lodrätt på skärmen. Man var tvungen att återgå till den gamla serverversionen. Varför slutade klienten att fungera?

Banken är missnöjd med att man var tvungen att gå tillbaka till den gamla versionen. Förutom att skriva menyn kors och tvärs vill man expandera utrikes och då måste klienten kunna visa menyvalen på olika språk. Dessutom har ledningen fått dyra råd från managementkonsulter som betonat vikten av att ha ett fält där man kan skriva välkomsthälsningar och tipsa om tjänster som fonder eller säkra sydeuropeiska placeringar.

Banken har skrivit ett 30-årigt dyrt smalbandsavtal och är missnöjd med att det skickas alltför mycket trafik (långa strängar) från servern till klienten. En mossig gammal assemblerprogrammerare i bankens styrelse har därför bestämt att alla trafikmeddelanden mellan server och klient högst får vara tio (10!) bytes.

Det enda undantaget när man får skicka mer data är när klientens programvara ska uppdateras. Banken har planer på att expandera utrikes och då måste klienten kunna visa menyvalen på olika språk. Dessutom vill man ha ett fält där man kan skriva välkomsthälsningar och tipsa om tjänster.

Ett par säkerhetskonsulter från McFinurley har skrivit en rapport där de förordar att programmet ska utökas så att man kan logga in på banken med kortnummer och kod. Men det är inte allt. Eftersom man är orolig för kortbedrägerier ska användaren bekräfta varje uttag med en särskild engångskod. Listor med engångskoder skickas ut till användaren separat.

Banken har nu beslutat sig att vända sig till dig. Du ska skriva om både klient- och serverdelen så att följande krav uppfylls.

Krav

Kommunikationen

1. Du ska specificera protokoll så att
 - (a) serverdelen kan uppdatera de meddelanden som visas i klienten
 - (b) Övriga meddelanden (login, transaktioner ...) är högst tio byte stora
2. Protokoll ska kommenteras och illustreras separat. Visa bild på data som ska skickas samt en eller flera scenarion med dataflödet.

Systemet

1. Man ska kunna logga in med kortnummer och sifferkod
2. Vid varje uttag ska en tvåsiffrig kod matas in och kontrolleras mot en till användaren utskickad kodlista. För att förenkla redovisningen ska samtliga listor hårdkodas till alla udda tal 01, 03, 05 ... 99
3. Banken ska kunna spara tillstånd för flera användare. Olika användare ska kunna logga in och ändra sitt saldo. Vid redovisningen ska minst tre personer finnas inlagda.
4. Menyn som skrivs ut ska innehålla ett fält på max 80 tecken där banken kan skriva välkomsthälsningar eller göra reklam för bankens tjänster (jämför en banner på en internetsajt).
5. Banken ska kunna byta meddelande på välkomsthälsningen (till godtycklig sträng) utan att starta om server eller klient.
6. Klientprogrammet ska kunna välja språk. Banken ska kunna erbjuda ett nytt språk utan att klienten behöver göras om.
7. Banken kräver att koden ska kommenteras bra!

Vid redovisning

1. ... ska du ha tre terminalfönster uppe med en server och två klienter som körs.
2. ... ska du visa en **illustration** över ditt protokoll. Även de enklaste protokoll godkänns bara de är bra illustrerade. Skilj på beskrivningen av datat och dataflödet.
3. ... ska protokolldokumentationen (illustration + kort beskrivning) vara **så bra att en godtycklig D-student ska kunna skriva en ny klient utan att behöva fråga om klargöranden.**
4. ... ska du kunna redogöra för olika tillstånd i klient och serverdelen.
5. ... ska du ha en editor uppe med all källkod.
6. ... ska du kunna redogöra för alla detaljer i koden. Det är tillåtet att skriva om hela koden.