ELEKTRONIK
TIDNINGEN

Robert Niemi
Androidexpert
Enea Wireless Solutions Center, Lund

Mobilbolagen älskar Android

Svenska Enea insåg tidigt värdet i Googles Linux-plattform Android. Företaget har etablerat sig med utbildning och partnertjänster för dig som vill testa Android i inbyggda system. Robert Niemi ger här en grundkurs i plattformens arkitektur, barnsjukdomar och potential.

Redaktör
Jan Tångring
jan@etn.se
0734-17 13 09

EMBEDDED
EXPERT

Mobilbolagen älskar Android

Men Googles iPhone-för-alla är inte riktigt klar ännu

Av Robert Niemi, Enea

Robert Niemi jobbar på Eneas Android Competence Center i Lund. Med sin gedigna erfarenhet av inbyggda realtidssystem är han en uppskattad lärare och mentor. Han är en ledande expert i Eneas Androidsatsning, som bygger på Eneas specialistkompetens inom integration av mobila plattformar och Linux. I satsningen ingår utbildning, workshops och specialisttjänster för att hjälpa kunder att komma igång med sina Androidprojekt.

Två saker skiljer Android från andra Linuxdistributioner: det är ett enanvändarsystem och det är anpassat för mobila plattformar. Organisationen Open Handset Alliance och dess 51 medlemmar ligger bakom konceptet. Huvudsponsorn Google tar en betydande roll.

Ledordet är öppenhet och det visar sig dels genom att det mesta av källkoden är öppen men främst genom att alla applikationer har lika värde. Om en användare inte tycker om webbläsaren kan den ersättas med en annan. Redan nu finns tusentals applikationer som tillför eller ersätter inbyggd funktionalitet.

Trots de nya möjligheterna för mobiltelefontillverkarna väntar en besvärlig balansgång och de har bråda dagar att så snabbt som möjligt få fram telefoner enligt det nygamla konceptet att åter placera användaren i centrum och samtidigt tillgodose operatörernas krav på säkerhet och funktionalitet.

Androidbaserade mobiltelefoner tillhör segmentet ”smartphones”. Medan en traditionell så kallad ”feature phone” ofta har ett realtidsoperativsystem som exempelvis Enea OSE i botten och all funktionalitet tillhandahållen av tillverkaren, har smartphones ett applikationsoperativsystem som Linux, Symbian eller Windows CE i botten och endast en del av applikationerna skrivna av tillverkaren. Övriga applikationer köps och laddas ner av användaren allt eftersom funktionaliteten efterfrågas. I denna aspekt är alltså en smartphone mer lik en vanlig pc.

Men Android skiljer sig markant från andra smartphones. Alla gränssnitt är öppna för dem som vill skriva applikationer. Tillverkaren kan inte ha privata gränssnitt till viss funktionalitet utan att bryta mot Androidkonceptet. Tvärtom måste all funktionalitet finnas tillgänglig för alla.

Alla applikationer i Android är skrivna i programspråket Java. De är därefter kompilerade i två steg, först med en vanlig Javakompilator och sedan med en dexkompilator. Dex står för Dalvik executable format. Den färdiga koden exekveras sedan i en virtuell maskin som heter Dalvik.

Dalvik benämns ofta slarvigt som en Javamaskin, men detta är alltså felaktigt då Dalvik inte alls kör Javas bytekod utan sin egen variant, så kallad dexkod. Den interna layouten skiljer sig också på flera punkter. Dalvik är en registermaskin, det vill säga som de flesta moderna cpu:er, medan Javamaskiner är stackbaserade. Dessutom körs varje applikation i sin egen instans av den virtuella maskinen medan Javamaskiner för mobila plattformar traditionellt kör i separata trådar i samma instans.

Att Java används som primärt programmeringsspråk för Android för mycket gott med sig. Java är idag världens populäraste programspråk och det finns väldigt många utvecklare som behärskar språket. Nackdelen är dock att merparten av dessa utvecklare saknar erfarenhet av inbyggda system, något som märks tydligt på den ojämna kvaliteten av program som finns att tillgå för Android.

Exekveringsmodellen för Dalvik, att varje applikation får sin egen instans, eller egen process i operativsystemet, öppnar upp för en mycket speciell lösning för att få ett robust system. Genom att kombinera detta med det faktum att Android är ett enanvändarsystem använder man sig av separata användar- och gruppidentiteter för varje applikation. Vidare sätts mycket restriktiva rättigheter och en applikation kan per automatik inte komma åt annat än sina egna resurser.

Det finns dock möjligheter att dela resurser. Detta styrs genom att användaren tillfrågas vid installation av applikationen om exempelvis den egna telefonboken skall vara tillgänglig eller om applikationen tillåts komma åt nätverket.

Inter Process Communication (IPC) är ett annat område där Android skiljer sig från andra Linuxbaserade system. De klassiska funktionerna för IPC är begränsade och är ersatta med en teknik som går under namnet Binder. Den kommer ursprungligen från ett gammalt OS som heter BeOS, men har även funnits i olika former för andra OS. Binder är i mångt och mycket en objektorienterad kommunikationskanal, likt COM eller CORBA. I Androids tappning är dock Binder mycket mindre generell och istället specialanpassad för just denna plattform.

Ett delsystem i Android som fått mycket publicitet är C-biblioteket som används för all native-kod. För att undvika typiska licensproblem som brukar drabba företag när de utvecklar produkter baserade på Linux har man gjort en egen implementation av standard C-biblioteket. Detta bibliotek har fått namnet Bionic C library. Det är skyddat av en BSD-licens istället för den GPL-licens som skyddar

många av de andra C-bibliotek som används i Linux. Det innebär att mobiltelefontillverkare kan lägga till funktionalitet i C-biblioteket utan att behöva publicera sin källkod.

Mycket annan kod som ingår i Androidkonceptet skyddas av den likartade Apachelicensen, men det finns även delar som skyddas av GPL eller LGPL. Myllret av licenser är svårt att överblicka och innebär risker för tillverkarna.

Bionic C library är på intet sätt revolutionerande men har ytterligare några andra intressanta egenskaper utöver licensen. Tanken är att biblioteket länkas statiskt till C-applikationen, vilket gör att varje process har sin egen kopia av hela biblioteket. Det betyder att koden måste vara kompakt. För att uppnå kravet har många specialfall tagits bort och endast nödvändig funktionalitet finns implementerad. Exempelvis är C++-supporten begränsad. Services och systemfunktionalitet måste helt enkelt skrivas i C.

FOTO MOTOROLA DROID

I oktober 2009 fanns 12 Androidmodeller i 32 mobilnät i 26 länder.

I Androidlägret brukade man inte se detta som något problem. Det var helt enkelt inte tanken att någon utom de själva skulle skriva kod på den här nivån. På senare tid har man dock insett att Java och Dalvik för med sig en del prestandaproblem och har därför publicerat ett NDK, Native Development Toolkit, för att möjliggöra användandet av C för tidskritiska delar av en applikation.

Ett klassiskt problemområde för många som utvecklar lite mer avancerade inbyggda system är filsystem. De bästa filsystemen för Linux kräver att blockenheten är av hårddisktyp. Så är inte fallet i smartphones och flera andra komplexa produkter. Istället behövs ett filsystem anpassat för en blockenhet av flashminnestyp. De vanligaste flashfilsystemen för Linux idag har antingen prestanda- eller stabilitetsproblem. Det rekommenderade filsystemet för Android heter Yaffs2 och där har man lyckats råda bot på många problem, men istället har man tillfört några andra. Exempelvis kommer inte Yaffs2 att kunna användas på moderna flashminnestyper utan rejäla utvecklarinsatser. I förlängningen kommer mobiltelefontillverkarna att välja andra filsystem, även om inget riktigt bra alternativ med öppen källkod finns i dagsläget.

En annan utmaning för smartphones är själva systemlayouten. De bygger ofta på ett System-on-chip (SoC) där flera cpu:er, som ofta kör flera olika operativsystem, måste kunna kommunicera på ett effektivt sätt. Android innehåller inga egentliga direktiv för hårdvaran, annat än minimikrav för minne och cpu-hastighet. En typisk layout innehåller en kraftfull applikationsprocessor som kör exempelvis Linux och en radioprocessor med ett robust realtidsoperativsystem, exempelvis Enea OSE. I konstellationen ingår även ofta en eller flera DSP:er som avlastar radioprocessorn med kodningen. DSP:en kör ofta ytterligare ett operativsystem, exempelvis Enea OSEck. För att få en pålitlig telefon måste kommunikationsvägarna mellan de olika processorerna vara robusta och effektiva. Eneas produkt Linx kan användas för att lösa dessa problem. Linx knyter enkelt samman exempelvis Linux med andra operativsystem. Linx för Linux är släppt med öppen källkod. Det finns även andra lösningar som lämpar sig väl.

En väldigt intressant aspekt med Android är att det börjar sprida sig till andra teknikområden än mobiltelefoner. Flera datortillverkare har exempelvis utlovat netbooks och mediaspelare som kör Android, och man börjar även se etablerade aktörer söka sig utanför sina vanliga marknader.

Det största problemet med Android är att det inte är en färdig plattform. Detta trots att det finns telefoner på marknaden och både Android och telefontillverkare har aggressiva releaseplaner. Exempelvis saknas fortfarande många vitala delar för native-utveckling. Det är inte heller klart hur tillverkarna skall kunna utöka plattformen för sina egna behov utan att bryta mot de publika gränssnitten.

Det går i dagsläget inte att lägga till finesser för exempelvis en mer avancerad kamera utan att först få detta godkänt av Androidalliansen (läs Google). Att produkten inte är färdig märks tydligt på den knapphändiga dokumentationen. Applikationslagret med alla sina Javagränssnitt är mycket bra dokumenterat, men för den som är intresserad av resten av plattformen finns endast dokumentation i form av källkod samt forum och bloggar. Det är förvisso bra med en stor community som är väldigt engagerad, men var och en som fördjupar sig i Android måste bilda sig en egen uppfattning om huruvida informationen man hittar är pålitlig.

Hårdvarukraven är också väldigt höga. Enligt specifikationen behövs åtminstone 128MB ram, 256MB flash och en cpu på minst 200MHz. Android är dock primärt tänkt som en plattform för high end-segmentet, vilket säkerligen gör att de flesta tillverkare känner sig tvingade att uppfylla minimikraven med god marginal.

Ytterligare ett problemområde är hur inkomsterna för applikationerna kommer att hanteras. Android följer samma modell som Apple skapat med sin App Store, men med öppenhet som ledord har man möjliggjort för flera olika kanaler att leverera applikationer till användarna. Google har sin egen programbutik, Android Market, där de tar inkomsterna från försäljningen. Men operatörer och telefontillverkare vill också vara med och kommer därför med all säkerhet att öppna sina egna programbutiker. Dessutom finns det möjlighet för både operatörer och tillverkare att begränsa öppenheten och tvinga användarna att endast använda deras butiker. Även om detta inte görs idag, kommer vi med all sannolikhet att se en förändring när Android börjar bli vanligt även i enklare telefonmodeller.

Slutligen finns säkerhetsaspekten. I och med att mobiltelefonen blir mer öppen och lik en PC ökar riskerna för virus och annan illvillig programkod. Vid installation av en applikation måste användaren godkänna vilka rättigheter applikationen ges, men sedan är det upp till utvecklaren av applikationen att inte missbruka rättigheterna.

Trots de problem som finns är Android det hetaste som hänt i mobilbranschen på länge och de flesta mobiltelefontillverkarna har stora projekt på gång. Det skapar också nya spännande utmaningar och möjligheter för tjänsteföretag och operatörer. ■