Erlang på Olimex SAM9-L9260

Per Söderstam, januari 2012

# Bakgrund

Under arbete med en routerliknande box för Emerson, hösten 2011, skojade vi till det och sade att ”detta skall man ju utveckla i Erlang”. Nu ville det sig inte bättre än att vi missade Emerson jobbet men iden om att köra Erlang på en liten Linuxnod fastnade i mitt huvud. Därför fick jag för mig att klämma in Erlang i den lilla SAM9:a som vi fick med oss hem från kursen i inbäddad Linux.

För att nå hela vägen fram ville jag inte bara ha Erlang på noden utan också en webserver. Nu vill det sig inte bättre än att det finns en webserver, Yaws, som är utvecklad i Erlang och som kör inbäddad i Erlang.

# Utvecklingsmiljö

Allt arbete på host är gjort från en Ubuntu distro som gick i VirtualBox på min vanliga arbetsmaskin. Värt att notera här, som en lite startpunkt, är att konfigurera VBox med direkt nätverksaccess så att du har en ordentlig LAN-adress.

Det är också ett tips att ge Linux i VBox:en en god storlek sin virtuella disk, 30+ GB blir bra då byggena kan ta en del plats.

# Procedur

Följande är en ungefärlig procedur, sammanfattad från alla de variationer och villospår jag tog under genomförandet av ”projektet”.

## Få tag på Erlang och Yaws

Det finns inget färdigt varför vi skall korskompilera från källkod. Programvarorna finns att hitta på

[www.erlang.org](http://www.erlang.org)

respektive

[yaws.hyber.org](http://yaws.hyber.org/).

Det spelar ingen större roll vilka versioner du väljer. Se bara till att de är hyffsat nya och stabila.

## …och Linux

För att få full funktion på Erlang och för att få Yaws att fungera över huvud taget behöver vi SSL och SSH biblioteken med associerade funktioner. Då vi korskompilerar behöver vi dessa kompilerade för SAM9:an också.

Jag valde i detta läge att bygga ett helt nytt Linuxsystem. Jag använde för detta ändamål Buildroot och byggde ett minimalt system grundat på 3.1.4 kärnan som innehåller SSL, SSH och crypto biblioteken förutom den vanliga påsen gott och blandat.

För att kunna jobba lätt och smidigt konfigurerar jag för NFS boot men undviker i övrigt så mycket annat som jag kan.

…att få ihop detta är en alldeles egen historia.

## Korskompilera Erlang

Erlang är relativt enkel att korskompilera när man väl fått till ett rotfilsystem. Det finns två varianter: den valiga configure/make och ett byggscript, otp\_build. Båda metoderna är väl dokumenterade, se INSTALL-CROSS.md. Jag använde otp\_build.

För att köra detta modifierar jag xcomp/erl-xcomp.conf.template för att peka ut rotfilträdet (erl\_xcomp\_sysroot), korskompileringsverktygen (erl\_xcomp\_build och erl\_xcomp\_host) )och sätta konfigureringsparametrar. I mitt fall stänger jag av Java, termcap, smp och hipe (erl\_xcom\_configure\_flags=’--without-java –without-termcap –disable-smp-support –disable-hipe’).

Därefter kör vi:

# ./otp\_build configure –xcomp-conf=<konfigureringsfilen>

för att konfigurera bygget. För att sedan bygga kör vi:

# ./otp\_build release <release directory>

Du bör nu ha fått ett minimalt system i <release directory>. Detta skall vara tillräckligt för att kunna köra Erlang.

## Installera Erlang

Börja med att gå till ditt <release directory> och kör

# ./Install –cross –minimal <target install directory>

där <target install directory> är absolutadressen till det bibliotek du vill placera Erlang i på target, t.ex /usr/local. Vad som händer är då att diverse vägar kodas in i byggets parameterfiler på byggmaskinen.

Packa sedan ihop <release directory> och packa upp det igen i <target install directory> i ditt rotfilsträd. Kör

# ./Install –cross –minimal <target install directory>

igen.

Presto! Nu borde du kunna starta ett Erlang skal på SAM9:an om alla övriga PATH:er och parametrar tillåter detta.

## Yaws

Yaws är en annan femma! Emedan Erlang är byggt med korskompilering i åtanke supportar Yaws inte kompilering till annat än Windows och några av de större Linux distributionerna, och explicit inte korskompilering.

Men det går ändå med lite handpåläggning.

## Korskompilera Yaws

Börja med att köra en normal ./configure med –host flaggan satt till din korskompilator. Detta borde resultera i att scriptet stannar med felkommentaren att det inte kan korskompilera testmodulerna. I detta felmeddelande borde det finnas en radreferens till konfigureringsscriptet.

Gå då till denna plats i scriptet. Där borde det finna en if-sats på omkring 20-40 rader som genererar detta felmeddelande. Försök ta bort hela denna sats! Kör ./configure igen, och om du fått bort rätt del borde konfigureringen gå igenom. Bygg.

## Installera Yaws

Gick detta bra är det bara att packa ihop, flytta till rotfilsystemet och packa upp på rätt plats (företrädesvis /usr/local om du inte angett annat till ./configure). Kom ihåg att utföra allt uppackande från SAM9:ans skal, som ju ser rotfilträdet ur rätt ”vinkel”.

Detta kommer dock inte att fungera med en gång. Under bygget blir Yaws alla konfigurationsfiler satta utifrån byggmaskinens vy. Plocka därför fram /usr/local/etc/yaws/yaws.conf och justera de inställningar som verkar felaktiga. Detta inkluderar diverse sökvägar och servernamn och parametrar.

Har du fått med alla inställningar borde Yaws dra igång och du får se en lokal kopia av Yaws hemsida om du http:ar in på ditt lilla Linuxkort.

Hepp!