

2. atskaite

Vienkāršas programmas izpilde uz LPC-2478-STK izstrādes plates

Teorētiskā daļa

Veidot programmas izpildei uz Olimex LPC-2478-STK izstrādes plates var izmantojot C programmēšanas valodu, kas spēj nodrošināt assembleram līdzīgu tiešo piekļuvi pie reģistru vērtībām. Vienkāršas programmas kods var izskatīties līdzīgi gan rakstot to izpildei uz galda datora, gan izpildei uz iestrādes plates. Atšķirības kodā parādās, ja programma tiek izmantotas darbības ar reģistriem un to vērtību uzstādīšana vai mainīšana. Programmu, kas tika nokompilēta, izmantojot standarta kompilatorus galda datoriem, nevar palaist uz iestrādes plates. Lai būtu iespējams uz galda datora kompilēt programmas, kuras tiks palaistas uz cita veida arhitektūras, ir jāinstalē un jānokonfigurē kroskompilators.

Kroskompilators (ang. cross compiler) ir kompilators, kurš viena tipa datorā ģenerē programmu, kura paredzēta izpildei cita tipa datorā. Tas nozīmē, ka, izmantojot attiecīgo kroskompilatoru, ir iespējams, piemēram, uz galda datora ar Intel X86 vai Intel X64 arhitektūru ģenerēt programmu, kas tiks palaista uz LPC-2478-STK izstrādes plates. Kroskompilators parasti tiek izmantots programmu kompilēšanai platformām, kur veikt kompilēšanu tieši uz ierīces nav iespējams. Tas attiecas uz mikrokontrolleriem un cita veida iegultām sistēmām, kuras neatbalsta operētājsistēmas un kompilatorus.

Kompilējot programmu izpildei uz citas arhitektūras, ir nepieciešams izmantot ne tikai kroskompilatoru, bet arī attiecīgas bibliotēkas. Bibliotēkas satur papildus konfigurācijas un kompilēšanas informāciju, kas ir nepieciešama veiksmīgai programmas ģenerēšanai. LPC-2478-STK plate darbojas ar uClinux operētājsistēmu, tāpēc, ģenerējot programmu, kroskompilatoram ir jāievēro gan izstrādes plates arhitektūras, gan operētājsistēmas īpašības.

Praktiskā daļa

Praktiskās daļas veikšanai tika izmantota pamācība (www.henjab.se). Darba gaita tika veidoti personīgi pieraksti un pamācības, kas ļāva likvidēt iepriekšminētās pamācības nepilnības un atvieglot tālāko darbu. Programmas kompilēšana un komunikācija ar plati tika veikti Ubuntu vidē.

Pirmais, kas tika izpildīts, bija kroskompilatora arm-linux-gcc instalēšana. Lai iegūtu kompilēšanai nepieciešamas bibliotēkas, sākumā bija nepieciešams uzbūvēt uClinux.

uClinux būvēšana (visas darbības tiek veiktas Ubuntu terminālā ar root tiesībām).

Darba direktorijas veidošana:

- **mkdir /opt/arm**
- **cd /opt/arm**

Nepieciešamo failu atarhivēšana (faila atrodas uClinux diskā):

- **tar -xvzf ../Utils/arm-linux-tools-20061213.tar.gz**
- **tar -xvzf ../uClinux/uClinux-dist-lpc_2478_stk-20081007.tgz**
- **wget http://www.uclinux.org/pub/uClinux/uclinux-elf-tools/gcc-3/uclinux-tools-20040603/genromfs-0.5.1.tar.gz**
- **tar -xvzf genromfs-0.5.1.tar.gz**
- **cd genromfs-0.5.1**
- **make**
- **cd ..**

Ceļu konfigurēšana:

- **export PATH=/opt/arm/usr/local/bin:/opt/arm/genromfs-0.5.1:\$PATH**
- **cd uClinux-dist-lpc_2478_stk**

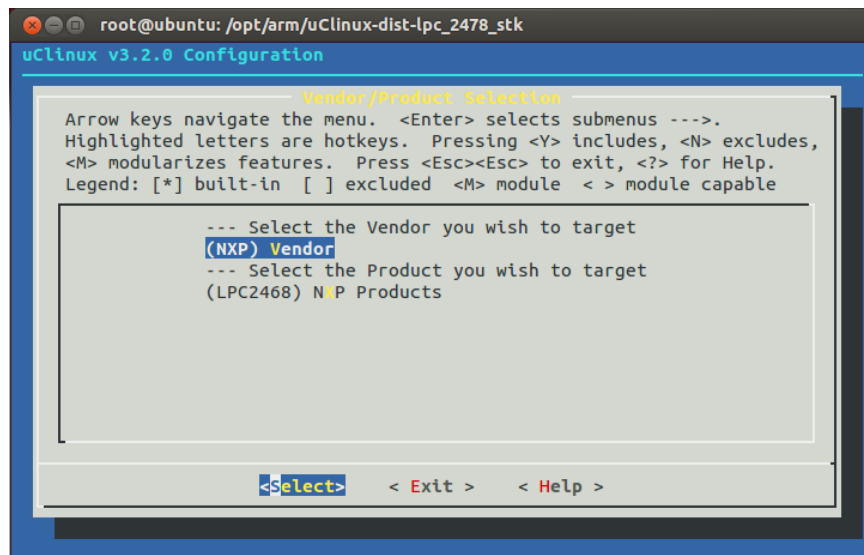
Direktorijas attīrīšana:

- **make distclean**
- **\rm -f tools/ucfront-g++**
- **\rm -f tools/ucfront-gcc**
- **\rm -f tools/ucfront-ld**
- **\rm -f tools/cksum**

Palaižam konfigurāciju izvēlni:

- **make menuconfig**

Izvēlnē ir nepieciešams nokonfigurēt izstrādes plates ražotāju un pašu plati: Vendor/Product Selection —> Vendor —> NXP (LPC2468 tiks izvēlēta pēc noklusējuma) (1.att.). Izejot no konfigurēšanas izvēlnes (Exit->Exit->Yes) ir jāpalaiž **make dep** un tad **make**.



1.att. Konfigurēšanas izvēlne

Ja kompilēšanas process noritēja veiksmīgi, tad apakšdirektorijā images ir jābūt failiem romfs_new.img un vmlinux_new.bin. Veicot šo darbu, kompilācijas soli notika vairākas kļūdas, kas sākumā nebija saprotamas, tomēr vēlāk tika noskaidrots, ka kļūdas ir radušas bibliotēku trūkuma dēļ.

Kad uClinux ir izveidots un kļūst pieejama ucfrent-gcc bibliotēka, var veikt programmas kompilēšanu. Laboratorijas darbā tika izmantota vienkārša programma, kas pēc palaišanas uz LPC-2478-STK izstrādes plate izvada Ubuntu terminālā paziņojumu „Hello world!”. Programmas kodu ir iespējams apskatīt zemāk.

```
#include <stdio.h>
int main(int argc, char** argv)
{
    printf("\r\nHello world!\r\n");
    return 0;
}
```

Programmas direktorija HelloWorld, kas satur helloworld.c un Makefile, tika ievietota /home/edo/ direktorijā. Pirms uzsākt programmas ģenerēšanu ir nepieciešams pārvietoties uz programmas direktoriju un nokonfigurēt arm-linux-gcc un ucfrent-gcc ceļus:

- **cd HelloWorld**
- **export PATH=/opt/arm/usr/local/bin:/opt/arm/genromfs-0.5.1:\$PATH**
- **export PATH=/opt/arm/uClinux-dist-lpc_2478_stk/tools:\$PATH**

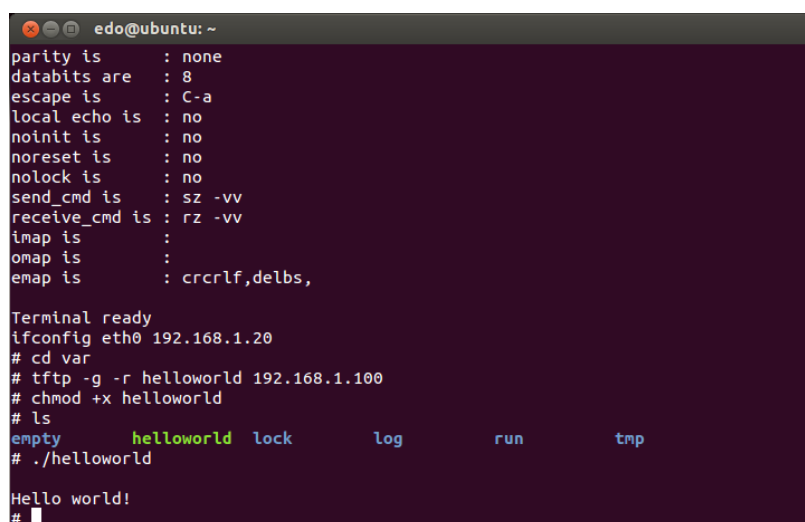
Pēc ceļu nokonfigurēšanas var palaist komandu **make**, kas veiks automātisku programmas ģenerēšanu saskaņā ar Makefile uzdotiem parametriem. Ja ģenerēšana notika pareizi, tad HelloWorld direktorijā ir jāparādās failam helloworld.

uClinux uz LPC-2478-STK var ielādēt, izmantojot vienu no iepriekšējā laboratorijas darba atskaitē aprakstītam metodēm. Kad uClinux ir ielādēts un palaists uz plātes, programmu var ielādēt plātē, lietojot TFTP serveri:

- **cd /var**
- **ifconfig eth0 192.168.1.20**
- **tftp -g -r helloworld 192.168.1.100**
- **chmod +x helloworld**
- **./helloworld**

Direktorija var ir mainīgo direktorija un vienīga, kur lietotājs var ievietot failus un palaist programmas. 192.168.1.20 ir nokonfigurēta izstrādes plates IP-adrese. 192.168.1.100 ir nokonfigurēta servera IP-adrese. Pirms palaišanas programmai jāatļauj izpildīties, izmantojot `chmod +x`.

Izpildot aprakstītās darbības, programma helloworld tika veiksmīgi izpildīta uz LPC-2478-STK izstrādes plates. Ubuntu terminālā izvade ir redzama attēlā (2.att.).



```
edo@ubuntu: ~
parity is      : none
databits are  : 8
escape is     : C-a
local echo is : no
noinit is     : no
noreset is    : no
nolock is     : no
send_cmd is   : SZ -vv
receive_cmd is : rz -vv
imap is       :
omap is       :
emap is       : crclrf,delbs,

Terminal ready
ifconfig eth0 192.168.1.20
# cd var
# tftp -g -r helloworld 192.168.1.100
# chmod +x helloworld
# ls
empty    helloworld lock    log     run     tmp
# ./helloworld
Hello world!
#
```

2.att. Programmas ielāde un palaišana

Secinājumi

Otrā laboratorijas darba gaitā tika nokonfigurēts arm-linux-gcc kroskompilators, uzbūvēts uClinux un ģenerēta vienkārša programma, kas vēlāk tika izpildīta uz LPC-2478-STK izstrādes plates.

uClinux būvēšana un pareizo kroskompilatora ceļu konfigurēšana bija grūtāka un laikietilpīgāka šī laboratorijas darba daļa, jo vairākas reizes notika kompilēšanas kļūdas. Vēlāk tika noskaidrots, ka Ubuntu nebija pieejamas dažas nepieciešamas kroskompilatoram pamata bibliotēkas, kuras bija jāmeklē, apskatot kompilēšanas kļūdas, un atsevišķi jāielādē.

Darba rezultātā ir iegūtas pamatzināšanas par kroskompilatora darbību un programmu ģenerēšanu izpildei uz LPC-2478-STK izstrādes plates.