Szymon Dyszewski

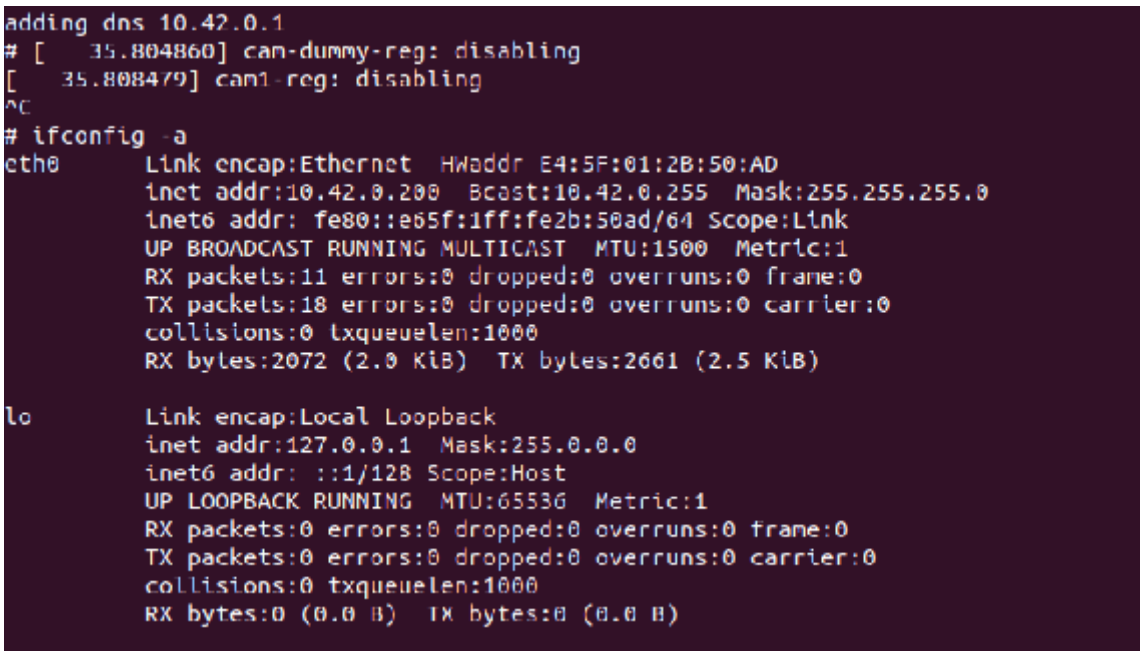
Laboratorium nr 1
SKPS

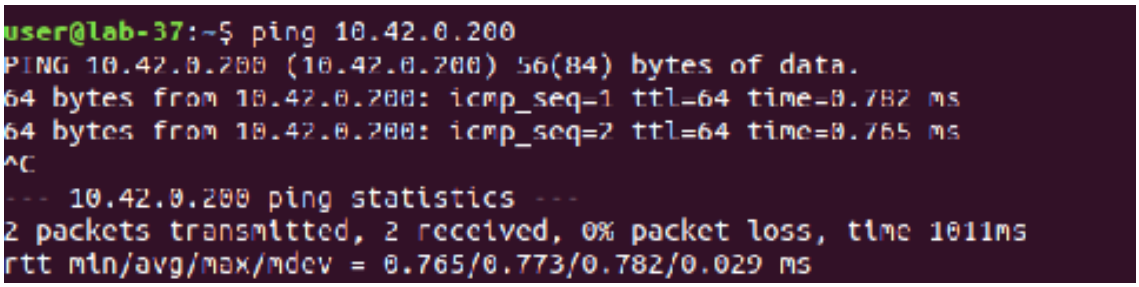
1. Przygotowanie środowiska.

Po otrzymaniu zestawu początkowego przeszedłem do składania zestawu. Zestaw został złożony i zatwierdzony przez prowadzącego. Wszystkie urządzenia i kable zostały ułożone w sposób ułatwiający pracę na komputerze.

2. Pierwsze uruchomienie RPi.

Uruchomienie zasilania spowodowało włączenie się urządzenia. Wpisanie komendy `tio /dev/ttyUSB0` skutkowało poprawnym włączeniem się tio. Otrzymałem informację connected. Pojawiły się logi z bootloadera i uzyskałem możliwość zalogowania się. Załadował się system ratunkowy Raspberry Pi OS. Wpisanie podanych danych logowania zakończyło się poprawnym wejściem. Sprawdziłem stan połączenia sieciowego na RPi poleceniem ifconfig oraz ping 10.42.0.1.





Szymon Dyszewski

3. Kopiowanie plików na RPi

Na komputerze host postawiłem serwer HTTP przy użyciu polecenia: python3 -m http.server. W katalogu roboczym stworzyłem plik test. Przy użyciu komendy wget plik został pobrany na urządzenie RPi.

```
user@lab-37:~$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.42.0.200 - - [26/Oct/2022 16:57:49] "GET /lesl HTTP/1.1" 200 -
```

```
# wget http://10.42.0.1:8000/lesl
--1970-01-01 00:07:49--  http://10.42.0.1:8000/lesl
Connecting to 10.42.0.1:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [application/octet-stream]
Saving to: 'test'

test                    [ <=>                ]       0  --.-KB/s    in 0s

1970-01-01 00:07:49 (0.00 B/s) - 'test' saved [0/0]
```

4. Kompilacja obrazu linuxa w BR

a) Obraz dla Raspberry Pi 4B z initramfs

Ściągnąłem plik buildroot-2021.08.tar.bz. Przy użyciu komendy tar -xf rozpakowałem jego zawartość w docelowym katalogu. Zbudowałem Buildroota wykonując polecenie make raspberrypi4_64_defconfig (domyślna konfiguracja). Następnie poleceniem make menuconfig zmieniłem konfigurację:
- Toolchain – external toolchain
- włączyłem initramfs
- wyłączyłem ext2/3/4
- włączyłem kompresję (gzip)

Po zmianach w konfiguracji zbudowałem obraz poleceniem make.

b) Obraz dla Raspberry Pi 4B bez initramfs

Na początku usunąłem stary obraz poleceniem make linux-dirclean. Ustawiłem w konfiguracji (poleceniem make menuconfig):
- Toolchain – external toolchain
- zmieniłem hostname na sdyszewski
- wyłączyłem initramfs
- włączyłem ext2/3/4
- włączyłem kompresję (gzip)

Ponownie zbudowałem obraz poleceniem make. Budowanie zakończono pomyślnie.

Szymon Dyszewski

5. Uruchomienie zbudowanego obrazu

a) Initramfs

Skopiowałem pliki Image, bcm2711-rpi-4-b.dtb, cmdline.txt z katalogu /output/images/. Na RPi ściągnąłem je przy użyciu komendy wget i skopiowałem komendą cp <file> /mnt/user/. Następnie przeszedłem do restartu RPi przy użyciu komendy reboot. Przytrzymując przycisk SW4 zbootowałem zbudowany system.

```
# wget http://10.42.0.1:8000/kernel8.img
--1970-01-01 00:44:17--  http://10.42.0.1:8000/kernel8.img
Connecting to 10.42.0.1:8000... failed: Connection refused.
# wget http://10.42.0.1:8000/kernel8.img
--1970-01-01 00:49:38--  http://10.42.0.1:8000/kernel8.img
Connecting to 10.42.0.1:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 47213056 (45M) [application/octet-stream]
Saving to: 'kernel8.img'

kernel8.img         100%[===================>]  45.03M  11.2MB/s    in 4.0s

1970-01-01 00:49:42 (11.2 MB/s) - 'kernel8.img' saved [47213056/47213056]

# wget http://10.42.0.1:8000/cmdline.txt
--1970-01-01 00:49:51--  http://10.42.0.1:8000/cmdline.txt
Connecting to 10.42.0.1:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 65 [text/plain]
Saving to: 'cmdline.txt'

cmdline.txt         100%[===================>]      65  --.-KB/s    in 0s

1970-01-01 00:49:51 (1.94 MB/s) - 'cmdline.txt' saved [65/65]

# wget http://10.42.0.1:8000/bcm2711-rpi-4-b.dtb
--1970-01-01 00:50:26--  http://10.42.0.1:8000/bcm2711-rpi-4-b.dtb
Connecting to 10.42.0.1:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 49749 (49K) [application/octet-stream]
Saving to: 'bcm2711-rpi-4-b.dtb'

bcm2711-rpi-4-b.dtb 100%[===================>]  48.58K  --.-KB/s    in 0.004s

1970-01-01 00:50:26 (13.1 MB/s) - 'bcm2711-rpi-4-b.dtb' saved [49749/49749]

# ls
adapter_test.sh      cmdline.txt          test
bcm2711-rpi-4-b.dtb  kernel8.img
# cp cmdline.txt /mnt/user/
# cp cmdline.txt /mnt/user/
# cp bcm2711-rpi-4-b.dtb  /mnt/user/
# cp kernel8.img  /mnt/user/
# reboot
```

Szymon Dyszewski

Utworzyłem plik tempfile poleceniem touch.

```
Welcome to Buildroot
buildroot login: root
# ls
# cd /mnt/user/
-sh: cd: can't cd to /mnt/user/: No such file or directory
# ls
# touch tempfile
# ls
tempfile
```

Zrebootwałem RPi i sprawdziłem zawartość katalogu domowego – plik nie przetrwał restartu.

```
Welcome to Buildroot
buildroot login: root
# ls
# ls
#
```

b) brak Initramfs

Przekopiowałem dodatkowy plik rootfs.ext2 z host na RPi oraz nagrałem system plików na partycji 2 karty SD dd if=rootfs.ext2 of=/dev/mmcblk0p2 bs=4096.Dodatkowo na 1 partycji wgrałem nowy Image, bcm2711-rpi-4-b.dtb, cmdline.txt.

```
user@lab-37:~$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.42.0.200 - - [26/Oct/2022 19:48:29] "GET /Image HTTP/1.1" 200 -
10.42.0.200 - - [26/Oct/2022 19:48:46] "GET /bcm2711-rpi-4-b.dtb HTTP/1.1" 200 -
10.42.0.200 - - [26/Oct/2022 19:48:55] "GET /cmdline.txt HTTP/1.1" 200 -
10.42.0.200 - - [26/Oct/2022 19:49:08] "GET /rootfs.ext2 HTTP/1.1" 200 -
```

```
# cp Image /mnt/user/
# cp bcm2711-rpi-4-b.dtb  /mnt/user/
# cp cmdline.txt   /mnt/user/
# dd if=rootfs.ext2 of=/dev/mmcblk0p2 bs=4096
32768+0 records in
32768+0 records out
```

Niestety system bootując się zawieszał się niezależnie od ustawień configa. Wspólnie z prowadzącym nie udało nam się znaleźć przyczyny. Zostały podjęte 3 próby, aż czas się skończył.

```
MESS:00:00:06.871117:0: brfs: File read: 65 bytes
MESS:00:00:06.875069:0: No compatible kernel found
MESS:00:00:06.877223:0: Device tree loaded to 0x2eff3800 (size 0xc7f8)
```