

[SKPS] Sprawozdanie z laboratorium nr 1

Zespół 104C – Jakub Rozkosz, Hubert Gołębiowski

Na początku podłączyliśmy się do konsoli UART za pomocą programu tio
Po przygotowaniu stanowiska uruchomiliśmy płytke i załączyliśmy system ratunkowy
Następnie uruchomiliśmy DHCP i sprawdziliśmy połączenie sieciowe

Adres płytki: 10.42.0.155

```
Welcome to Buildroot rescue OS
rescue login: root
# [ 35.804898] cam-dummy-reg: disabling
[ 35.808518] cam1-reg: disabling

# udhcpc
udhcpc: started, v1.33.1
udhcpc: sending discover
udhcpc: sending select for 10.42.0.155
udhcpc: lease of 10.42.0.155 obtained, lease time 3600
deleting routers
adding dns 10.42.0.1
# ifconfig
eth0      Link encap:Ethernet  HWaddr E4:5F:01:2B:50:80
          inet addr:10.42.0.155  Bcast:10.42.0.255  Mask:255.255.255.0
          inet6 addr: fe80::e65f:1ff:fe2b:5080/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0
          TX packets:19 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2072 (2.0 KiB)  TX bytes:2731 (2.6 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

# ping 10.42.0.1
PING 10.42.0.1 (10.42.0.1): 56 data bytes
64 bytes from 10.42.0.1: seq=0 ttl=64 time=0.963 ms
64 bytes from 10.42.0.1: seq=1 ttl=64 time=0.878 ms
64 bytes from 10.42.0.1: seq=2 ttl=64 time=0.872 ms
64 bytes from 10.42.0.1: seq=3 ttl=64 time=0.850 ms
^C
--- 10.42.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.850/0.890/0.963 ms
#
```

następnie uruchomiliśmy buildroota
make raspberrypi4_64_defconfig
make menuconfig

Konfiguracja systemu z initramfs w buildroocie:

Toolchain → Toolchain type → zmieniamy z **Buildroot toolchain** na **External toolchain**

Filesystem images → zmieniamy z **ext2/3/4 root filesystem** na **cpio the root filesystem** oraz włączamy kompresję **gzip**

System configuration → **System hostname** ustawiamy na **Rozkosz_Golebiowski**

kompilujemy system komendą `make`

po skompilowaniu przechodzimy do katalogu `/output` i stawiamy serwer http za pomocą komendy `python3 -m http.server`

następnie w konsoli płytki pobieramy 3 pliki za pomocą komend:

```
wget http://10.42.0.1:8000/images/Image
```

```
wget http://10.42.0.1:8000/images/bcm2711-rpi-4-b.dtb
```

```
wget http://10.42.0.1:8000/images/rpi-firmware/cmdline.txt
```

zmieniamy nazwę pliku Image na kernel8.img:

```
mv Image kernel8.img
```

montujemy partycję 1:

```
mount /dev/mmcblk0p1 /mnt
```

kopiujemy 3 powyższe pliki do katalogu `/mnt/user`:

```
cp kernel8.img ../mnt/user
```

```
cp bcm2711-rpi-4-b.dtb ../mnt/user
```

```
cp rpi-firmware/cmdline.txt ../mnt/user
```

i uruchamiamy ponownie płytkę:

```
reboot
```

tym razem przy uruchamianiu trzymamy przycisk SW4

Niestety z niewiadomo jakich przyczyn nie udało się uruchomić obrazu linuxa.

Po udanym wykonaniu drugiej części laboratorów przystąpiliśmy do powtórzenia części pierwszej – dokładnie przeszliśmy przez wszystkie kroki, upewniając się, że wszystkie pliki zostały prawidłowo wygenerowane i że znajdują się w odpowiednim miejscu. Po reboocie natomiast odpaliła się wersja bez initramfs, gdyż pliki pozostawały po wykonaniu reboota. Po dyskusji z prowadzącym nie udało się dojść do przyczyny.

Konfiguracja systemu bez initramfs w buildroocie:

(użyliśmy konfiguracji z poprzedniej części jako bazy, którą modyfikowaliśmy)

Filesystem images → zmieniamy z **cpio the root filesystem** na **ext2/3/4 root filesystem**, włączamy kompresję **gzip** i zmieniamy rozmiar pliku ext2 na 200M

ponawiamy proces kopiowania plików na płytke, tym razem przesyłamy także plik **rootfs.ext2**

wgrywamy system plików:

```
dd if=rootfs.ext2 of=/dev/mmcblk0p2 bs=4096
```

i uruchamiamy płytke z wciśniętym przyciskiem SW4

tworzymy plik testowy komendą **touch /test.txt**

```
Welcome to Buildroot
Rozkosz_Golebiowski login: root
# ls
# cd ..
# ls
bin      lib      lost+found  opt      run      tmp
dev      lib64    media       proc     sbin     usr
etc      linuxrc  mnt         root     sys      var
# cd root
# touch /test.txt
# ls
# cd ..
# cd root
# ls
# pwd
/root
# cd ..
# ls
bin      lib64    mnt      run      tmp
dev      linuxrc  opt      sbin     usr
etc      lost+found  proc    sys      var
lib      media    root     test.txt
# reboot
```

uruchamiamy ponownie, aby sprawdzić czy plik się zachował

```
Welcome to Buildroot
Rozkosz_Golebiowski login: root
# ls
# cd ..
# ls
bin      lib64    mnt      run      tmp
dev      linuxrc  opt      sbin     usr
etc      lost+found  proc    sys      var
lib      media    root     test.txt
# ponownie odpalone i wciaz znajduje sie plik
```