

## Dodatek do zadań projektowych z przedmiotu Programowanie systemów wbudowanych.

### 1. Instalacja pakietów niezbędnych do przeprowadzenia kompilacji skróśnej jądra OS Linux

```
$ sudo apt-get install git cmake make binutils build-essential libncurses5-dev bc bison flex libssl-dev
```

### 2. Pobranie narzędzi do kompilacji skróśnej w systemie hosta (np. do sciezka\_katalogu)

```
$ cd sciezka_katalogu  
$ git clone https://github.com/raspberrypi/tools
```

### 3. Edycja pliku .bashrc użytkownika

Dopisz do zmiennej środowiskowej \$PATH ścieżkę do narzędzi do kompilacji skróśnej. Jeśli pracujesz na maszynie 32-bitowej:

```
$ echo PATH=$PATH:~/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabi-hf-raspbian/bin >> ~/.bashrc  
$ source ~/.bashrc
```

Jeśli pracujesz na maszynie 64-bitowej:

```
$ echo PATH=$PATH:~/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabi-hf-raspbian-x64/bin >> ~/.bashrc  
$ source ~/.bashrc
```

### 4. Pobranie źródeł jądra OS Linux (np. do sciezka\_linux)

```
$ cd sciezka_linux  
$ git clone --depth=1 https://github.com/raspberrypi/linux
```

### 5. Kompilacja jądra OS Linux

Utworzenie zmiennej środowiskowej KERNEL:

```
$ KERNEL=kernel7
```

Konfiguracja (zawartość bcm2709\_defconfig zostanie skopiowana do .config):

```
$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-hf- bcm2709_defconfig
```

Uruchomienie kompilacji (opcja all spowoduje wygenerowanie obrazu jądra, drzewa urządzeń oraz modułów jądra):

```
$ make -j n ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-hf- all
```

Symbol n oznacza liczbę procesorów (rdzeni procesora) w systemie hosta.

### 6. Przygotowanie systemu plików na karcie SD

Przed instalacją przygotowanych plików należy zadbać, by na karcie SD znajdował się odpowiedni system plików. W tym celu można nagrać na kartę SD obraz systemu Raspbian, pobrany ze strony: <https://www.raspberrypi.org/downloads/raspbian/>

Kopiowanie obrazu systemu Raspbian na kartę SD (np. widoczną w systemie jako /dev/sdb):

```
$ sudo dd if=plik_raspbian.img of=/dev/sdb
```

Moduły jądra zapisane są w katalogu `/lib/modules`. Przed instalacją przygotowanych samodzielnie plików można usunąć zawartość tego katalogu:

```
$ sudo rm -r /lib/modules/*
```

## 7. Instalacja plików w systemie plików na karcie SD

W katalogu, w którym wykonana została kompilacja jądra systemu Linux, utwórz katalog `mnt`, a w nim katalogi: `fat32` i `ext4`. Następnie zamontuj w nich obie partycje karty SD:

```
$ mkdir mnt
$ mkdir mnt/fat32
$ mkdir mnt/ext4
$ sudo mount /dev/sdb1 mnt/fat32
$ sudo mount /dev/sdb2 mnt/ext4
```

Instalacja modułów jądra:

```
$ sudo make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-
INSTALL_MOD_PATH=mnt/ext4 modules_install
```

Jeśli podczas kompilacji modułów jądra napotkasz problem z rozpoznawaniem ścieżki do kompilatora skrótnego (wyeksportowanej w zmiennej środowiskowej `PATH`), spróbuj zalogować się na maszynie hosta jak `root`, ponownie wyeksportować ścieżkę do kompilatora skrótnego w zmiennej `PATH` (tym razem w pliku `.bashrc` w katalogu `/root/`) oraz powtórzyć polecenie.

Kopiowanie obrazu jądra oraz drzewa urządzeń:

```
$ KERNEL=kernel7
$ cp mnt/fat32/$KERNEL.img mnt/fat32/$KERNEL-backup.img
scripts/mkknimg arch/arm/boot/zImage mnt/fat32/$KERNEL.img
$ cp arch/arm/boot/dts/*.dtb mnt/fat32/
$ cp arch/arm/boot/dts/overlays/*.dtb* mnt/fat32/overlays/
$ cp arch/arm/boot/dts/overlays/README mnt/fat32/overlays/
```

Odmontowanie systemu plików:

```
$ umount mnt/fat32
$ umount mnt/ext4
```