C-Programme für Raspberry Pi compilieren

Toolchain (verschiedene C-Compiler, die ARM-Code erzeugen) für Raspi per `git` herunterladen:

```
git clone --depth=1 https://github.com/raspberrypi/tools.git
```

`depth=1` bedeutet, dass nur die NEUSTE Version des aktuellen Zweigs heruntergeladen wird, nicht die ganze Historie. Das kann eine Menge Platz sparen.

Einbinden des Compilers in den Suchpfad:

```
PATH="/home/benutzer/tools/arm-bcm2708/gcc-linaro-arm-linux-
gnueabihf-raspbian/bin:$PATH"
```

Nun kann der C-Compiler aufgerufen werden:

```
arm-linux-gnueabihf-gcc -static -o hello hello.c
```

Das „file"-Utility zeigt an, dass es sich um ein statisch gelinktes ARM-Binary handelt:

```
file hello
hello: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV),
statically linked, for GNU/Linux 2.6.26,
BuildID[sha1]=d5e53061d000f07ec73ea5d9a34c7af807af45c7, not
stripped
```

Um den Linux-Kernel zu compilieren, genügt es, ein paar Environment-Variablen zu setzen und „make" im Kernel-Source-Verzeichnis aufzurufen:

```
export ARCH=arm
export CROSS_COMPILE=/media/mmcblk0p1/Pi/rpi-kernel/tools/arm-
bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/bin/arm-linux-
gnueabihf-
export INSTALL_MOD_PATH=/media/mmcblk0p1/Pi/rpi-kernel/rt-kernel
export INSTALL_DTBS_PATH=/media/mmcblk0p1/Pi/rpi-kernel/rt-kernel
export KERNEL=kernel7
```

Jetzt ins Linux-Kernel-Source-Verzeichnis wechseln, und:

```
make zImage
make modules
make dtbs
make modules_install
make dtbs_install
```

Die ersten beiden „`make`" dauern je nach System etwa 4 Stunden!