**INSTITUTO POLITÉCNICO NACIONAL**

**UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS**

**AVANZADAS**

**Asignatura:** Sistemas Operativos en Tiempo Real

**Profesor: M**aza Casas Lamberto

**Grupo:** 3MV9

**Compilador Cruzado**

**Alumno: T**éllez Pérez Luis José

Pasos para la construcción del compilador Cruzado:

1 cd ~

2 mkdir gcc\_all && cd gcc\_all

Let’s download the software that we’ll use for building the cross compiler:

1 wget https://ftpmirror.gnu.org/binutils/binutils-2.28.tar.bz2

2 wget https://ftpmirror.gnu.org/gcc/gcc-6.3.0/gcc-6.3.0.tar.gz

3 wget https://ftpmirror.gnu.org/glibc/glibc-2.24.tar.bz2

4 wget https://ftpmirror.gnu.org/gcc/gcc-8.1.0/gcc-8.1.0.tar.gz

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

Next, extract the archives and erase them:

1 tar xf binutils-2.28.tar.bz2

2 tar xf glibc-2.24.tar.bz2

3 tar xf gcc-6.3.0.tar.gz

4 tar xf gcc-8.1.0.tar.gz

5 rm \*.tar.\*

GCC also needs some prerequisites which we can download inside the source folder:

1 cd ..

2 cd gcc-6.3.0

3 ./contrib/download\_prerequisites

4 rm \*.tar.\*

5 cd ..

6 cd gcc-8.1.0

7 ./contrib/download\_prerequisites

1. rm \*.tar.\*

Next, create a folder in which we’ll put the cross compiler and add it to the path:

1 cd ..

2 sudo mkdir -p /opt/cross-pi-gcc

3 sudo chown $USER /opt/cross-pi-gcc

4 export PATH=/opt/cross-pi-gcc/bin:$PATH

Copy the kernel headers in the above folder, see Raspbian [documentation](https://www.raspberrypi.org/documentation/linux/kernel/building.md) for more info about the kernel:

1 cd ..

2 cd linux

3 KERNEL=kernel7-

4 make ARCH=arm INSTALL\_HDR\_PATH=/opt/cross-pi-gcc/arm-linux-gnueabihf headers\_install

Next, let’s build Binutils:

1 cd ..

2 mkdir build-binutils && cd build-binutils

3 ../binutils-2.28/configure --prefix=/opt/cross-pi-gcc --target=arm-linux-gnueabihf --with-arch=armv6 --with-fpu=vfp --with-float=hard --disable-multilib

4 make -j 8

5 make install

Open in a text editor ubsan.c from gcc-6.3.0/gcc/, find line 1474:

1 || xloc.file == '\0' || xloc.file[0] == '\xff'

and change it to:

//1 || xloc.file[0] == '\0' || xloc.file[0] == '\xff'

save and close the file.

GCC and Glibc are interdependent, you can’t fully build one without the other, so we are going to do a partial build of GCC, a partial build of Glibc and finally build GCC and Glibc. You can read more about this in [Preshing’s article](http://preshing.com/20141119/how-to-build-a-gcc-cross-compiler/).

1 cd ..

2 mkdir build-gcc && cd build-gcc

3 ../gcc-6.3.0/configure --prefix=/opt/cross-pi-gcc --target=arm-linux-gnueabihf --enable-languages=c,c++,fortran --with-arch=armv6 --with-fpu=vfp --with-float=hard --disable-multilib

4 make -j8 all-gcc

5 make install-gcc

Siguiendo el procedimiento en Cygwin:











