Cross Compiler and Installing the QEMU tool

Rajes manna
2021BITE063

October 9, 2023

1 Cross Compilation

Cross compilation is the process of compiling software on one platform (the host) to run on a different platform (the target). It is commonly used when you want to develop software for an architecture or platform different from the one you are developing.

2 Installing the ARM Cross Compiler

To begin cross-compiling for ARM, we need to install the `arm-linux-gnueabi` cross-compiler. This compiler allows us to generate executable code running on ARM-based systems. Below are the steps to install the compiler:

```
# Update package manager's package list
$ sudo apt update

# Install the ARM cross compiler
$ sudo apt install gcc-arm-linux-gnueabi

# Verify the cross compiler
$ arm-linux-gnueabi-gcc --version
```

3 Generate the assembly code and an executable for the hello-world.c

Creating a simple C program named ”hello-world.c” that prints the ”Hello, World!” message when executed, After that the ARM Linux cross-compiler (arm-linux-gnueabi-gcc) to compiles the C source code into both assembly code and an executable binary.

```c
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

```
# Generate assembly code
$ arm-linux-gnueabi-gcc -S -o hello-world.s hello-world.c

# Generate executable
$ arm-linux-gnueabi-gcc -o hello-world hello-world.c

```

4 Using QEMU for Testing

QEMU is a versatile emulator that allows us to run programs for different architectures on our host system. It is a valuable tool for testing and debugging code meant for ARM-based systems. Here's how to use QEMU to run ARM programs:

```
# Install QEMU if not already installed
$ sudo apt install qemu-system-arm

# Run an ARM program using QEMU
$ qemu-arm -L /usr/arm-linux-gnueabi/ ./hello-world
```