



NC STATE UNIVERSITY

Message Board

Course: CSC501 - 001 Forum: PA1 Discussion

Go to: [Different Course](#) | [Forum List](#) | [Topic List](#)

Author **Topic: MIPS to x86**

adhanas Sun Jan 19 18:18

This is a more generic Xinu question and not related to PA1. The question might be naive: How exactly is Xinu's assembly code converted to machine code?

I mean, the assembly code for interrupt handling (and maybe for few other modules) is for MIPS architecture. But, when I dis-assemble the corresponding machine code, I see x86 architecture specific code. So, the assembler does this.

But, how does the assembler ('as' in our case, I guess) know that the input assembly code is in MIPS architecture? In other words, where exactly is the cross compiling done?

I couldn't get anything in this regard by looking at the Makefile.

```
INCLUDE = -I../h
CFLAGS = -march=i586 -fno-builtin -c -Wall -O0 ${DEFS}
${INCLUDE}
SDEFS = -I../h
```

```
CC = /usr/bin/gcc
cPP = /usr/bin/gcc -E
AS = /usr/bin/as
```

```
intr.o: ../sys/intr.S
${CPP} ${SDEFS} ../sys/intr.S | ${AS} -o intr.o
```

hcnguye3 Sun Jan 19 22:14

I believe that we are using x86 version. (CFLAGS = -march=i586). I can't really find anything which is written for MIPS architecture in our version.

There is a XINU version written for MIPS arch, which can be found online.

Post New Topic

Post a Reply

WolfWare - Message Board - last update 30-Aug-2001