

# 2010 R&E: Computer System Education & Research

## Lab 7. High-Level Code to Machine Code

Using the following C program, go through the compilation steps from preprocessing to linking. After each step, print out the outcome. You should run it on x86-based Linux first. Then, use the MIPS cross-compiler to do the same thing.

Note: When it comes to object codes (object code after assembling and executable after linking), dump the object code with `objdump` (for x86 code) and `mips-elf-objdump` (for MIPS code) and print out the outcome.

```
#define min(x,y) ((x) < (y) ? (x) : (y));

int main()
{
    int a, b, c, d, e, f;

    a = min(b, c);
    d = min(e, f);

    return (min(a,d));
}
```

1. Let's go through each step of compilation

- `$> cpp test.c > test.i`
- `$> gcc -Wall -S test.i // check out the output file (test.s)`
- `$> as test.s -o test.o`

2. Let's just compile the code with `gcc` and observe the assembly code it generates

- `$> gcc -g test.c -o test`
- `$> objdump -S -D test.exe > test.dump`
  - Open 'test.dump' and observe how each sentence in the C code is translated into.