ΕΠΛ323 - Θεωρία και Πρακτική Μεταγλωττιστών

Lecture 2

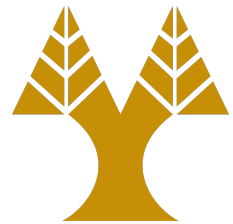
The Preprocessor

Elias Athanasopoulos
eliasathan@cs.ucy.ac.cy

Preprocessor

- Collects and prepares the code to be compiled
  - Code may be split in several files
  - **Cross-compiling:** certain code may be irrelevant for a particular architecture or Operating System
  - **Special features:** turn on/off certain compiler internal features
- Most compilers support pre-processing

Directives



- The pre-processor can be controlled using certain commands (or directives)
  - `#include, #define, #undef`
  - `#if, #else, #endif`
  - `#ifdef, #ifndef`
  - `#line, #error, #pragma`
- All directives start with: `'#'`

#define

```
#define MAX 100

…
if (x > MAX) {
   …
}
…
```

Faster!

```
…
if (x >100) {
   …
}
…
```

```
int MAX = 100;
…
if (x > MAX) {
   …
}
…
```

#ifdef, #ifndef

```
…
#ifdef DEBUG
printf( “Hello1: %d\n”, flag );
#endif
…
```

gcc -DDEBUG

```
…
printf( “Hello1: %d\n”, flag );
…
```

gcc

```
...
```

#ifdef #else #endif

```
…
#ifdef DEBUG
printf( “Hello1: %d\n”, flag );
#else
printf( “Hello2\n” );
#endif
…
```

gcc -DDEBUG

```
…
printf( “Hello1: %d\n”, flag );
…
```

gcc

```
…
printf( “Hello2\n” );
…
```

#include

hello.c

```
#include “hello.h”
…
main() {
}
…
```

hello.h

```
…
#ifndef _HELLO_H_
#define _HELLO_H_

int hello(int, int);

#endif
…
```

hello.c

```
…
#ifndef _HELLO_H_
#define _HELLO_H_

int hello(int, int);

#endif
…
main() {
}
…
```

hello.c

```
…

int hello(int, int);
…
main() {
}
…
```

This is what is going to be compiled

#if, #else, #elif, #endif

```c
#define A 100

#if A > 100
#define FOO 1
#else
#define FOO -1
#endif

#include <stdio.h>

int main(int argc, char *argv[]) {

    fprintf(stderr, "%d\n", FOO);

    return 1;
}
```

-1

#line, #error

```
#define A 100

#if A > 100
#define FOO 1
#else
#line 444
#error “A simple error”
#endif
            ...
```

```
pre-test.c:444:2: error: ”A simple error"
#error ”A simple error"
```

#pragma

- Implementation specific
- Used to tweak compiler’s internal features
  - #pragma pack(1), disables structure alignment in Microsoft Visual C