Assembly Programming

Lecture 07

Josh Brake
Harvey Mudd College

Outline

- Compilation process overview
- C to assembly examples
  - Arithmetic
  - Logical
  - Conditional execution
  - Loops
- Design Example

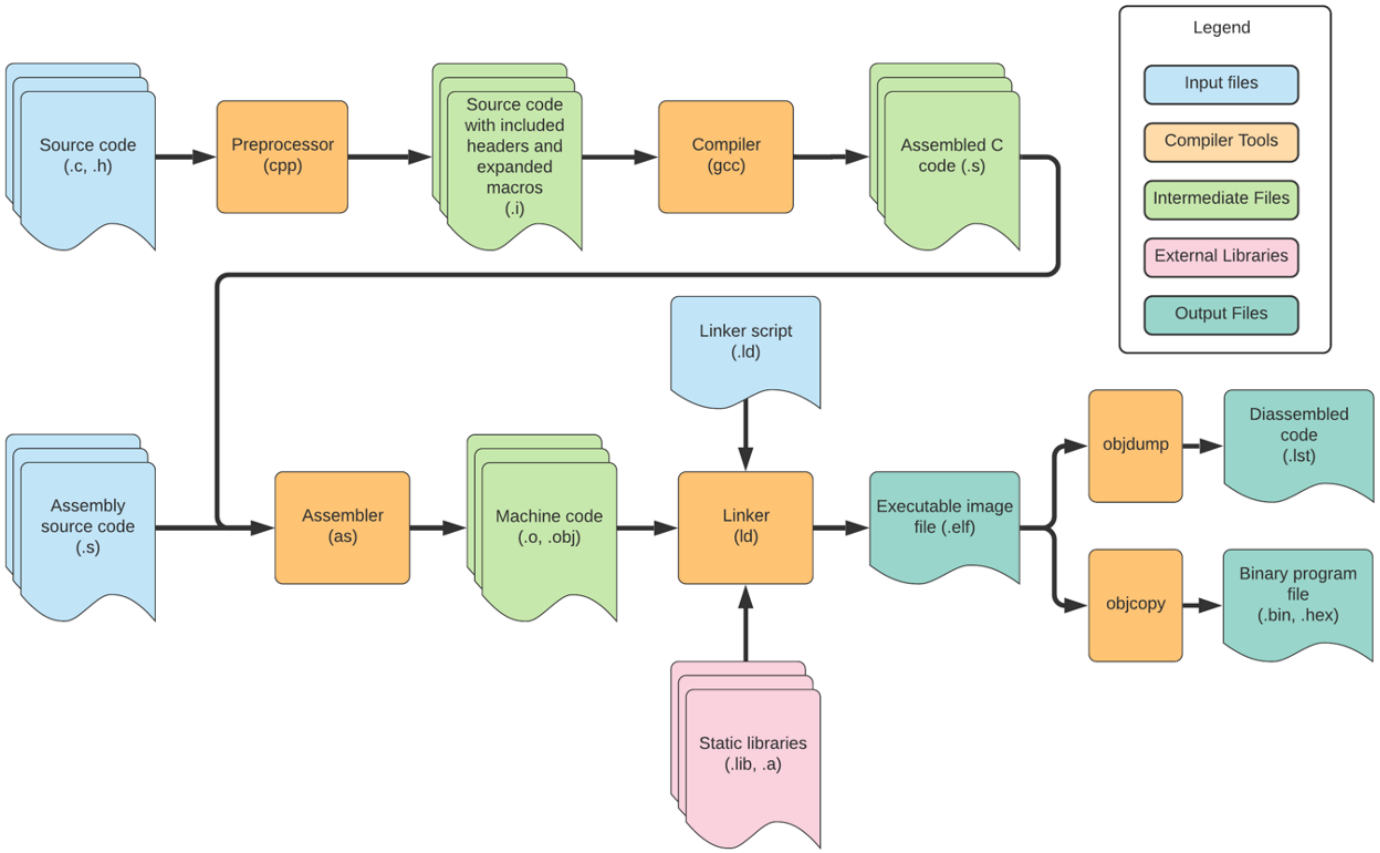
Learning Objectives

By the end of this lecture you should be able to…

- List the steps of the program compilation process
- Recall the assembly idioms for common C programming structures

Compilation Process

This example is for the GNU Compiler Collection (gcc)



C to Assembly Examples

Arithmetic Ex. 1

C

```
1  a = b + c;
```

ARM Assembly

Arithmetic Ex. 2

C

```c
a = b + 2 * c - d;
```

ARM Assembly

Arithmetic Ex. 3

C

```
1  a = d / 4;
```

ARM Assembly

Logical Ex. 1

C

```
1  a = b & c;
```

ARM Assembly

Logical Ex. 2

C

```
1  a = b | c;
```

ARM Assembly

Logical Ex. 3

C

```
1  a = b ^ c;
```

ARM Assembly

Logical Ex. 4

C

```
1  a = b << c;
```

ARM Assembly

Logical Ex. 5

C

```
1  a = b << c;
```

ARM Assembly

Conditional Execution Ex. 1

C

```c
if (a) b = 1;
```

ARM Assembly

Conditional Execution Ex. 2

C

```
1  if (a != b) c = d;
```

ARM Assembly

Conditional Execution Ex. 3

C

```c
if (a) c = 3;
```

ARM Assembly

Conditional Execution Ex. 4

C

```
1 if (a > b) {
2   // do stuff 1
3   }
4 else {
5   // do stuff 2
6   }
```

ARM Assembly

Conditional Execution Ex. 5

C

```c
if (a > b) c = 1;
else c = 0;
```

ARM Assembly

Loops Ex. 1

C

```c
int sum = 0, i = 0;
// sum in R0, i in R1

sum = 0;
for (i = 0; i < 10; i++)
    sum = sum + i;
```

ARM Assembly

Loops Ex. 2

C

```c
int i, j; // in R1, R2
int q; // in R3

for (i = 2; i < 8; i++)
  for (j = 1; j < i; j++)
    q = q + i - j;
```

ARM Assembly

Loops Ex. 3

C

```c
int i = 0; // in R1
unsigned int a1[20], a2[20];
// in R4, R5
for (i = 0; i < 20; i++){
  a1[i] = a2[i]/2;
}
```

ARM Assembly

Loops Ex. 4

C

```c
i = 1;
j = 0;
while (i <= 2048){
  a1[j++] = i;
  i = i * 2;
}
```

ARM Assembly

Loops Ex. 5

C

```c
char * str1, str2;
// R4, R5
int i = 0;

do {
  str2[i] = str1[i];
}
while (str1[i++]);
```

ARM Assembly

Design Example: Low-pass Filter

Problem Statement

Design a 4-sample running average filter for the following data

```
x = [42,54,60,72,78,86,100,112,124,130]
```

Steps

1. Write C code
2. Translate to assembly

C Code

Assembly Code

Assembly Code

Wrap up

- Assembly programming is most straightforward when you have a particular construct in a higher-level language like C in mind.
- Pay special attention to details like variable types (signed vs. unsigned), sizes, and the addressing modes (e.g., byte vs. word).
- Basic flow is load data into registers from memory, do something with the loaded data, store the result back in memory.