



컴파일러개론

WEEK10

안상준 | 202002514 | 2024.11.14

```
// Source code is decompiled from a .class file
public class Main {
    public Main() {
    }

    public static int fibo(int var0) {
        if (var0 <= 0) {
            return 0;
        } else if (var0 == 1) {
            return 1;
        } else {
            int var1 = 0;
            int var2 = 1;
            int var3 = 2;

            int var4;
            do {
                var4 = var1 + var2;
                var1 = var2;
                var2 = var4;
                ++var3;
            } while (var3 <= var0);

            return var4;
        }
    }

    public static void main(String[] var0) {
        System.out.println(fibo(10));
    }
}
```

```
● × sangjun19 ~/Desktop/Compiler/실습/10/jasmin-2.4 main ± java -jar jasmin.jar Main.j
Generated: Main.class
```

jvm assembly 코드를 jasmin 으로 class 파일로 변환한 결과이다.

```

.class public Main
.super java/lang/Object
.method public <init>()V
  aload_0
  invokespecial java/lang/Object/<init>()V
  return
.end method

.method public static fibo(I)I
  .limit stack 32
  .limit locals 32
  iload_0
  ifgt L1
  iconst_0
  ireturn
L1:
  iload_0
  iconst_1
  if_icmpne L2
  iconst_1
  ireturn
L2:
  iconst_0
  istore_1
  iconst_1
  istore_2
  iconst_2
  istore_3
Loop:
  iload_1
  iload_2
  iadd
  istore_4
  iload_2
  istore_1
  iload 4
  istore_2
  iinc 3 1
  iload_3
  iload_0
  if_icmple Loop
  iload_0
  if_icmple Loop
  iload_2
  ireturn
.end method

.method public static main([Ljava/lang/String;)V
  .limit stack 32
  .limit locals 32
  getstatic java/lang/System/out Ljava/io/PrintStream;
  bipush 10
  invokestatic Main/fibo(I)I
  invokevirtual java/io/PrintStream/println(I)V
  return
.end method

```

작성한 jvm assembly 코드이다. 처음 class 의 이름을 선언해 주고 생성자를 작성해 주었다. 그 다음 fibo 라는 함수를 선언해 주었다. 매개변수는 하나만 받고 리턴값은 하나이므로 (I)I 를 사용해 주었다. 매개변수로 받은 값을 iload_0 을 통해 스택에 push 하여 ifgt 조건문으로 0 보다 클 경우 L1 으로 이동하고 아닌 경우 0 을 리턴해 주었다. 그 다음 다시 로컬변수 슬롯에서 파라미터를 가져와 1 이 아닌 경우 점프하고 1 인 경우 1 을 리턴하였다. iconst 를 통해 상수 값을 istore 로 로컬변수 슬롯에 저장해 주었다. 0 은 파라미터가 저장돼있어 1 부터 3 까지 각 변수 a, b, count 에 0, 1, 2 의 값을 저장해 주었다. Loop 를 이용하여 반복문을 사용하였다. a, b 변수를 가져와 더하고 istore 를 통해 temp 라는 변수에 결과값을 저장해 주었다. iload_2 를 통해 b 의 값을 istore_1 로 a 에 대입시켜 주었다. iload 4 로 temp 의 값을 가져와 istore_2 로 b 에 저장해 주었다. iinc 3 1 은 변수슬롯 3 에 있는 값을 1 증가시키므로 count 변수가 1 증가하게 된다. iload_3, iload_0 으로 count, 파라미터 n 의 값을 가져와 count 가 큰 경우 b 의 값을 가져와 리턴하게 되고 아닌 경우 Loop 로 jump 하게 된다.

main 함수에서는 bipush 로 상수값 10 을 stack 에 push 하고 fibo 함수를 호출한다. 결과값을 출력하고 리턴하게 된다.

느낀점

코드를 보고 jvm assembly 코드로 변환하는 과정이 생각보다 어려웠다. 또한 작성한 코드가 jasmin 으로 class 파일로 변환하려고 할때 오류가 많아 힘들었다. 하지만 이러한 과정을 통해 코드가 컴파일러를 통해 어떻게 변환되는지 알 수 있었다.