



2023_compiler

JAVA BYTECODE 다뤄보기 (2)

202102699 정민경

(01 분반)



1. 수기로 작성한 JVM assembly 코드를 Jasmin 을 사용해 .class 파일로 변환 후 실행해보기.

```
(base) jeongmingyeong@jeongmingyeong-ui-MacBookAir ~/Downloads/jasmin-2.4 ls | grep Main.class
(base) x jeongmingyeong@jeongmingyeong-ui-MacBookAir ~/Downloads/jasmin-2.4 java -jar jasmin.jar /Users/jeongmingyeong/CNU/ITC_23/hw11/Main.j
Generated: Main.class
(base) jeongmingyeong@jeongmingyeong-ui-MacBookAir ~/Downloads/jasmin-2.4 ls | grep Main.class
Main.class
(base) jeongmingyeong@jeongmingyeong-ui-MacBookAir ~/Downloads/jasmin-2.4 java Main
5050
```

[수행과정]

1. Main.j 파일에 JVM assembly 코드 작성
2. jasmin 을 사용해서 Main.j 파일을 .class 파일로 변환
3. java Main 명령어를 사용해 실행결과 출력
 - ⇒ 1 ~ 100 까지의 합을 출력하는 코드이기 때문에 "5050" 이라는 int 값 출력 확인.

1-1. Main.j (JVM assembly) 코드 설명

- 각 코드 별 설명에는 과제를 해결하면서 사용한 JVM assembly 와 java code 가 함께 첨부되어있음.)
- Class name 및 생성자 선언

```
1  .class public Main
2  .super java/lang/Object
3
4  ; standard initializer
5  .method public <init>()V
6      aload_0
7      invokevirtual java/lang/Object/<init>()V
8      return
9  .end method
10
```

그림 1. JVM assembly

```
3  ✓ public static int sum(int n) {
4      int result = 0;
5      int i = 1;
6
7      while (i <= n) {
8          result += i;
9          i++;
10     }
11
12     return result;
13 }
```

그림 2. java code

Class name 은 Main 이고, Object class 상속받아서 object class 의 생성자를 호출하는 코드로 구현함.

- Public static int sum(int n) 함수 정의

```
10
11 .method public static sum(I)I
12 .limit stack 32
13 .limit locals 32
14
15 ; int result = 0
16 iconst_0
17 istore_1 ; result
18
19 ; int i = 1
20 iconst_1
21 istore_2 ; i
22
23 ; start loop (while)
24 Loop:
25 iload_2 ; load i
26 iload_0 ; load arg
27 if_icmpgt End ; if i > n then goto End label
28 iload_1 ; load result
29 iload_2 ; load i
30 iadd ; tmp = result + i
31 istore_1 ; result = tmp
32 iload_2 ; load i
33 bipush 1 ; push 1
34 iadd ; tmp = i + 1
35 istore_2 ; i = tmp
36 ldc 0 ; load 0
37 ifeq Loop ; if 0 == 0 then goto Loop
38 End:
39 iload_1 ; load result
40 ireturn ; return result
41 .end method
```

그림 3. JVM assembly

```
3  ✓ public static int sum(int n) {
4      int result = 0;
5      int i = 1;
6
7      while (i <= n) {
8          result += i;
9          i++;
10     }
11
12     return result;
13 }
```

그림 4. java code

Main class 의 sum 이라는 method 구현.

- Method sum 은 int type 의 인자 하나 받고, int type return.
- Sum 함수 안에서 반복문이 진행되기 때문에 2개의 label 표시
 2. Loop : 반복문의 body
 3. End : 반복 statement 이후 statement
(과제 코드에서는 return result)

Java 에서의 "+=" operator 는 assembly 형태로 표현 시 $a = a + b$ 의 꼴로 표현되기 때문에 JVM assembly 로 표현시에도 result 와 1 를 load 해 stack 에 push 한 후 두개를 iadd 로 더해서 다시 result 변수에 store 하는 것으로 구현함.

- Main 함수 정의

```
43 .method public static main([Ljava/lang/String;)V
44 .limit stack 32
45 .limit locals 32
46 bipush 100
47 istore_2 ; int n = 100
48 getstatic java/lang/System/out Ljava/io/PrintStream;
49 iload_2
50 invokestatic Main/sum(I)I
51 invokevirtual java/io/PrintStream/println(I)V
52 return
53 .end method
```

그림 5. JVM assembly

```
15 public static void main(String[] args) {
16     int n = 100;
17     System.out.println(sum(n));
18 }
19 }
```

그림 6. java code

Main class 의 main 함수에 대해 구현함.

과제 조건에서 1부터 100까지의 합 즉, sum 함수 호출 시 인자를 100으로 넘겨주는 것이 조건이었으므로, 100을 어떠한 변수 n 에 저장 후 sum 함수 호출 직전 n의 값 load 를 통해 sum(n) 을 호출해 결과값을 stack 에 저장 후 println 함수의 인자로 sum(n) 의 값을 주어 출력할 수 있도록 하였다.

Main 함수는 return type 이 void 이므로 return 만 진행해 main 함수 정의를 종료했다.

2. 후기

- 과제 난이도 : 약간 쉬움
- 후기 : hw10 을 잘 해결했고, 실습자료에 첨부해주신 Wikipedia 링크를 참고했다면 쉽게 해결할 수 있는 수준의 과제였던 것 같습니다.