

Architettura degli Elaboratori 2

Esercizi di gruppo validi come esonero per la parte pratica dell'esame

ESERCIZIO 3: Analisi del comportamento di metodi ricorsivi

INTRODUZIONE

L'**INVOKEVIRTUAL** è un'istruzione su 3 byte: il cui codice (relativo al nostro microprogramma) è **0xB6**, gli altri due byte insieme formano un offset che indica in quale word a partire dalla **CPP** è salvato l'indirizzo di base del metodo da richiamare nell'**area dei metodi**. Il byte a cui punta **CPP**+(offset x 4) e quello successivo compongono il numero di parametri + 1 (**OBJREF**) e i due successivi il numero di variabili. Dopo questi primi quattro byte dalla base del metodo nell'**area dei metodi** sarà presente la prima microistruzione da eseguire.

$0x08 \times 0x04 + 0x10000 (CPP) = 0x10020$

Byte Address (HEX)	Content (HEX)	Mnemonic	Labels
24
25	b6	invokevirtual	
26	0	torrihanoi_1	
27	8	torrihanoi_2	
28
2f
30	0	#param_1	
31	4	#param_2	
32	0	#var_1	
33	1	#var_2	
34	15	iload	
35
10001f
10020	0	torrihanoi_1	
10021	0	torrihanoi_2	
10022	0	torrihanoi_3	
10023	30	torrihanoi_4	

(Img. 1: Rappresentazione di parte dell'area dei metodi e constants pool corrispettiva alla compilazione del file TorriHanoi.jas)

Ad ogni chiamata della **INVOKEVIRTUAL** viene creato sullo stack un nuovo **record di attivazione** ovvero uno spazio in memoria per i parametri e le variabili locali del metodo chiamato (dato dai primi quattro byte alla base del metodo nell'**area dei metodi**). Inoltre saranno anche presenti i valori di **LV** e **PC** al momento della call per riuscire a ripristinarli all'esecuzione della **IRETURN**.

Alla base del record di attivazione (**LV**) si troverà il **link pointer** in cui è salvato l'indirizzo, all'interno del record, del **PC** da assegnare alla fine dell'esecuzione dello stesso. Successivamente troviamo i **parametri** e le **variabili**. In seguito troveremo il valore di **PC** ed **LV** al termine della **IRETURN**. A questo punto lo **SP** punterà all'indirizzo successivo a questi ultimi.

Word Address (HEX)	Content(hex)	Pointers
...
801c	0000006e	#4
801d	0000800f	#4
801e	00008023	#5
801f	00000003	#5
8020	00000002	#5
8021	00000001	#5
8022	00000003	#5
8023	00000054	#5
8024	00008017	#5
8025	0000802a	<-- LV #6
8026	00000002	#6
8027	00000002	#6
8028	00000003	#6
8029	00000001	#6
802a	00000054	#6
802b	0000801e	#6
802c	00000002	<-- SP #6

Byte Address (HEX)	Content (HEX)	Mnemonic	Labels
...
51	b6	invokevirtual	
52	0	torrihanoi_1	
53	8	torrihanoi_2	
54	13	ldc_w	

(Img. 2: Rappresentazione di parte della ram e area dei metodi corrispettiva al sesto livello di chiamata ricorsiva alla funzione **torriHanoi(n,a,b)** del file torriHanoi.jas)

IL METODO ASSEGNATO

Il metodo è `torriHanoi(n,a,b)` (line 45, file *TorriHanoi.jas*). La prima istruzione si trova codificata all'indirizzo `0x34` dell'area dei metodi. L'indirizzo `0x30` fa sempre parte del metodo ma serve solo alla `INVOKEVIRTUAL` per poter inizializzare correttamente lo stack.

Esso viene sempre richiamato nella linea 34 del file che corrisponde all'indirizzo `0x25` dell'**area dei metodi** e potrebbe venir chiamato (non con una certezza assoluta) alle linee 67 e 82 del file che corrispondono agli indirizzi `0x51` e `0x6b`. Questi ultimi fanno parte della ricorsione all'interno del metodo.

Nelle zone dell'**area dei metodi** evidenziate in **azzurro** troviamo: il caricamento sullo stack dei parametri e la chiamata all'`INVOKEVIRTUAL` con i successivi due byte che compongono l'offset dalla **CPP** (8).

Byte Address (HEX)	Content (HEX)	Mnemonic	Labels
20
21	10	bipush	
22	1	byte	
23	10	bipush	
24	3	byte	
25	b6	invokevirtual	
26	0	torrihanoi_1	
27	8	torrihanoi_2	
28
...
30	0	#param_1	
31	4	#param_2	
32	0	#var_1	
33	1	#var_2	
34	15	iload	
35	1	n	
36	10	bipush	
37	1	byte	
38	9f	if_icmpeq	
39	0	sposta_1	
3a	38	sposta_2	
3b	10	bipush	
3c	6	byte	
3d	15	iload	
3e	2	a	
3f	64	isub	
40	15	iload	
41	3	b	
42	64	isub	
43	36	istore	
44	4	c	
45	13	ldc_w	
46	0	objref_1	
47	0	objref_2	
48	15	iload	
49	1	n	
4a	10	bipush	
4b	1	byte	
4c	64	isub	
4d	15	iload	
4e	2	a	
4f	15	iload	
50	4	c	
51	b6	invokevirtual	
52	0	torrihanoi_1	
53	8	torrihanoi_2	
54	13	ldc_w	
55	0	objref_1	
56	0	objref_2	
57	15	iload	
58	2	a	
59	15	iload	
5a	3	b	
5b	b6	invokevirtual	
5c	0	sposta_1	
5d	9	sposta_2	
5e	60	iadd	
5f	13	ldc_w	
60	0	objref_1	
61	0	objref_2	
62	15	iload	
63	1	n	
64	10	bipush	
65	1	byte	
66	64	isub	
67	15	iload	
68	4	c	
69	15	iload	
6a	3	b	
6b	b6	invokevirtual	
6c	0	torrihanoi_1	
6d	8	torrihanoi_2	
6e	60	iadd	
6f	ac	ireturn	
70	13	ldc_w	sposta
71	0	objref_1	
72	0	objref_2	
73	15	iload	
74	2	a	
75	15	iload	
76	3	b	
77	b6	invokevirtual	
78	0	sposta_1	
79	9	sposta_2	
7a	ac	ireturn	
7b

(Img. 2: Rappresentazione di parte dell'area dei metodi corrispettiva alla compilazione del file *TorriHanoi.jas*)

ANALISI DEL COMPORTAMENTO DEL METODO RICORSIVO