

Hjörvar Sigurðsson

The diagram illustrates the execution of a program with nested forking and printing. The sequence of operations is as follows:

- Initial Process:** Executes `print` (outputs 0) and `fork`.
- First Forked Process:** Executes `print` (outputs 0) and `fork`.
- Second Forked Process:** Executes `print` (outputs 0) and `fork`.
- Third Forked Process:** Executes `print` (outputs 1) and `fork`.
- Fourth Forked Process:** Executes `print` (outputs 1) and `fork`.
- Fifth Forked Process:** Executes `print` (outputs 1) and `fork`.
- Sixth Forked Process:** Executes `print` (outputs 1) and `fork`.
- Seventh Forked Process:** Executes `print` (outputs 1) and `fork`.
- Eighth Forked Process:** Executes `print` (outputs 1) and `fork`.
- Ninth Forked Process:** Executes `print` (outputs 1) and `fork`.
- Tenth Forked Process:** Executes `print` (outputs 1) and `fork`.
- Eleventh Forked Process:** Executes `print` (outputs 1) and `fork`.
- Twelfth Forked Process:** Executes `print` (outputs 1) and `fork`.
- Thirteenth Forked Process:** Executes `print` (outputs 1) and `fork`.
- Fourteenth Forked Process:** Executes `print` (outputs 1) and `fork`.
- Fifteenth Forked Process:** Executes `print` (outputs 1) and `fork`.
- Sixteenth Forked Process:** Executes `print` (outputs 1) and `fork`.
- Seventeenth Forked Process:** Executes `print` (outputs 1) and `fork`.
- Eighteenth Forked Process:** Executes `print` (outputs 1) and `fork`.
- Nineteenth Forked Process:** Executes `print` (outputs 1) and `fork`.
- Twentieth Forked Process:** Executes `print` (outputs 1) and `fork`.

The diagram shows that the output values are determined by the sequence of forking and printing operations. The output values are 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1.

Fjöldi 0-tákna sem prentast út verður $2^k - 1$ þar sem í hverri lykkju tvöfaldast fjöldi ferla en sökum þess að 0-táknið er prentað fyrir fork()-skipunina þá verður 0-tákn ekki prentað í einu ferli (má sjá á mynd 1).

c. Það þarf að geyma i) gistu. Það er vegna þess að í gistum eru geymd mikilvæg minnisvistföng fyrir hvert ferli.

d. Það afritar vistfangsrými og alla opna skráarbenda foreldrisins.

4.

a.

		TLBT								TLBI					
		0x7...								0x0...					
Bit position		13	12	11	10	9	8	7	6	5	4	3	2	1	0
VA =	0x073	0	0	0	1	1	1	0	0	1	1	1	0	1	1
		VPN								VPO					
		0x1C								0x3B					

MMU ber saman VPN (0x1C) sýndarvistfangsins við TLB til að athuga hvort afrit af VPN sé í skyndiminninu. Það notar TLBI til að ákvarða Set, og TLBT til að ákvarða Tag.

Set	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid
0	03	—	0	09	0D	1	00	—	0	07	02	1
1	03	2D	1	02	—	0	04	—	0	0A	—	0
2	02	—	0	08	—	0	06	—	0	03	—	0
3	07	—	0	03	0D	1	0A	34	1	02	—	0

(a) TLB: 4 sets, 16 entries, 4-way set associative

Í TLB er færsla þar sem Set = TLBI = 0, Tag = TLBT = 07, og Valid = 1, þannig að það er smellur og við tökum raunsíðunúmerið PPN = 02. Raunsíðunúmerið er sett saman við VPO til að mynda raunvistfangið PA (m.ö.o., VPO færist beint yfir).

		CT						CI				CO	
		0x02						0xE				0x03	
Bit position		11	10	9	8	7	6	5	4	3	2	1	0
PA =	0xBB	0	0	0	0	1	0	1	1	1	0	1	1
		PPN						PPO					
		0x02						0x3B					

MMU ber saman CT (tag), CI (idx), og CO (blk 0-3) við skyndiminnið:

Idx	Tag	Valid	Blk 0	Blk 1	Blk 2	Blk 3
0	19	1	99	11	23	11
1	15	0	—	—	—	—
2	1B	1	00	02	04	08
3	36	0	—	—	—	—
4	32	1	43	6D	8F	09
5	0D	1	36	72	F0	1D
6	31	0	—	—	—	—
7	16	1	11	C2	DF	03
8	24	1	3A	00	51	89
9	2D	0	—	—	—	—
A	2D	1	93	15	DA	3B
B	0B	0	—	—	—	—
C	12	0	—	—	—	—
D	16	1	04	96	34	15
E	<u>13</u>	1	83	77	1B	D3
F	14	0	—	—	—	—

(c) Cache: 16 sets, 4-byte blocks, direct mapped

CT okkar er 0x02 en í skyndiminninu er tagið fyrir index E 0xC, eða 13.
Því er skyndiminnisskellur.

- b. Þar sem VPO ræður PPO sem svo ræður CI og CO, þá bý ég til VA sem hefur VPO sem leiðir til CI og CO sem ekki eru í skyndiminninu – t.d. VPO = 011000, en þá er CI 6 og CO 0, en það leiðir til skyndiminnisskells.
Þar sem VPN ræður TLBT og TLBI, þá læt ég VPN hluta VA vera þannig að TLBT og TLBI leiði til TLB smells – t.d. VPN = 00001101, en þá er TLBT 3 og TLBI 1, en það leiðir til TLB smells.

		TLBT						TLBI							
		0x3						0x1							
Bit position		13	12	11	10	9	8	7	6	5	4	3	2	1	0
VA =	0x358	0	0	0	0	1	1	0	1	0	1	1	0	0	0
VPN										VPO					
0xD										0x18					

		CT						CI				CO	
		0x2D						0x6				0x0	
Bit position		11	10	9	8	7	6	5	4	3	2	1	0
PA =	0xB58	1	0	1	1	0	1	0	1	1	0	0	0
		PPN						PPO					
		0x2D						0x18					

c. Já, það er hægt.

Vistfang sem varpast á þann hátt að mengið er 0 og merkið 0x03:

		TLBT						TLBI							
		0x3						0x0							
Bit position		13	12	11	10	9	8	7	6	5	4	3	2	1	0
VA =	0x318	0	0	0	0	1	1	0	0	0	1	1	0	0	0

Vistfang sem varpast á þann hátt að mengið er 1 og merkið er 0x03:

		TLBT						TLBI							
		0x3						0x1							
Bit position		13	12	11	10	9	8	7	6	5	4	3	2	1	0
VA =	0x358	0	0	0	0	1	1	0	1	0	1	1	0	0	0

5.

a.

$$N = 2^{16} \Rightarrow n = 16$$

$$M = 2^{10} \Rightarrow m = 10$$

$$P = 16 = 2^p \Rightarrow p = 4$$

$$VPO = \log_2(P) = 4$$

$$PPO = VPO = 4$$

$$\#VPN \text{ Bits} = \#Virtual \text{ Address Bits} - \#Page \text{ Offset bits}$$

$$\#VPN \text{ Bits} = 16 - 4 = 12$$

$$\#PPN \text{ Bits} = \#Physical \text{ Address Bits} - \#Page \text{ Offset bits}$$

$$\#PPN \text{ Bits} = 10 - 4 = 6$$

$$TLB \text{ hefur } 2 \text{ sets} = 2^t = 2^1 \Rightarrow t = 1.$$

$$\#TLBI \text{ Bits} = 1$$

$$\#TLBT = \#VPN \text{ Bits} - \#TLBI \text{ Bits} = 12 - 1 = 11$$

	TLBT											TLBI				
Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VA =																
VPN												VPO				

Mynd 5.1. Skipting sýndarvistfangs.

Bit position	9	8	7	6	5	4	3	2	1	0
PA =										
PPN						PPO				

Mynd 5.2. Skipting raunvistfangs.

b.

	TLBT											TLBI				
	0x137											0x0				
Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VA = 0x26E7	0	0	1	0	0	1	1	0	1	1	1	0	0	1	1	1
VPN												VPO				
0x26E												0x7				

Mynd 5.3. Sýndarvistfangið 0x26E7.

TLB

Set	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid
0	137	37	1									
1												

Mynd 5.4. TLB sem inniheldur gildi fyrir sýndarvistfangið 0x26E7.

c.

- i. Sýndarvistföng þar sem bitinn í 4. sæti er 1, en þá er TLBI 0x1.
- ii. Það er pláss fyrir fjórar færslur í mengi 1 í þessu TLB.

Þar sem TLBI bitinn þarf að vera 1, en TLBT bitarnir geta í raun tekið hvaða gildi (þó það kemur aðeins smellur ef TLBT samsvarar tagi í færslu í TLB með valid-gildið 1) þá eru 2^{11} mögulegar útgáfur af sýndarvistfanginu sem geta varpast í mengi 1 í þessu TLB.

- d. Já, ef að PPN fyrir þetta sýndarvistfang breytist þá breytist raunvistfangið.