

Worksheet 8

Name: _____

UID: _____

1.

- a. The program below has a bug. The thread is supposed to sleep for 1 second and then print a string. However, when we run it on our system, nothing prints. Why?
- b. You can fix this bug by replacing the `exit` function in line 10 with one of two different `Pthread` function calls. Which ones?

```
#include "csapp.h"
void *thread(void *vargp):
int main(){

    pthread_t tid;
    Pthread_create(&tid,NULL,thread, NULL);
    exit(0);
}

/*Thread routine*/
void *thread(void *vargp){
    Sleep(1);
    printf("Hello, world!\n");
    return NULL;
}
```

1.

Given a 32 bit virtual address space and a 24-bit physical address, determine the number of bits in the VPN, VPO, PPN, and PPO for the following page sizes P:

P	VPN bits	VPO bits	PPN bits	PPO bits
1 KB				
2 KB				
4 KB				
8 KB				

2.

For this question we will look at the translation of virtual addresses to physical addresses.

- Memory is byte addressable and memory accesses are to 1 byte words
 - Virtual addresses are 14 bits
 - Physical addresses are 12 bits
 - The page size is 64 bytes
 - The TLB is 4-way set associative with 16 entries
 - The cache is a direct mapped with a 4-byte line size with 16 lines
- a. Which bits of the virtual address indicate the following:
 - VPO (Virtual Page Offset)
 - VPN (Virtual Page Number)
 - TLBI (TLB Index)
 - TLBT (TLB Tag)
 - b. Which bits of the physical address indicate the following:
 - PPO (Physical Page Offset)
 - PPN (Physical Page Number)
 - CO (Cache Block Offset)
 - CI (Cache Index)
 - CT (Cache Tag)
 - c. For the virtual address 0xD95 and the following TLB and Cache states below...
 - i. Identify the VPO, VPN, TLBI, TLBT.
 - ii. What is the corresponding physical address? Identify the PPO, PPN, CO, CI, and CT.
 - iii. What was the value returned?

Set	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid
0	15		0	00	16	1	12		0	0F		0
1	11		0	03		0	07	3D	1	13		0
2	0D	2E	1	03	04	1	09		0	0A		0
3	04		0	3B		0	09		0	11	12	1

Index	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3
0	3D	1	3F	A2	42	00
1	21	0				
2	1F	1	4A	61	1D	EE
3	12	1	22	3C	2F	54
4	13	1	9B	5A	B5	01
5	2E	1	77	69	C4	82
6	11	1	10	11	FF	48
7	19	0				
8	0D	1	8C	6F	25	A7
9	0B	0				
A	3D	0				
B	16	1	5A	24	B3	51
C	32	1	D3	00	10	20
D	22	0				
E	24	1	52	17	7F	A9
F	04	0				

3.

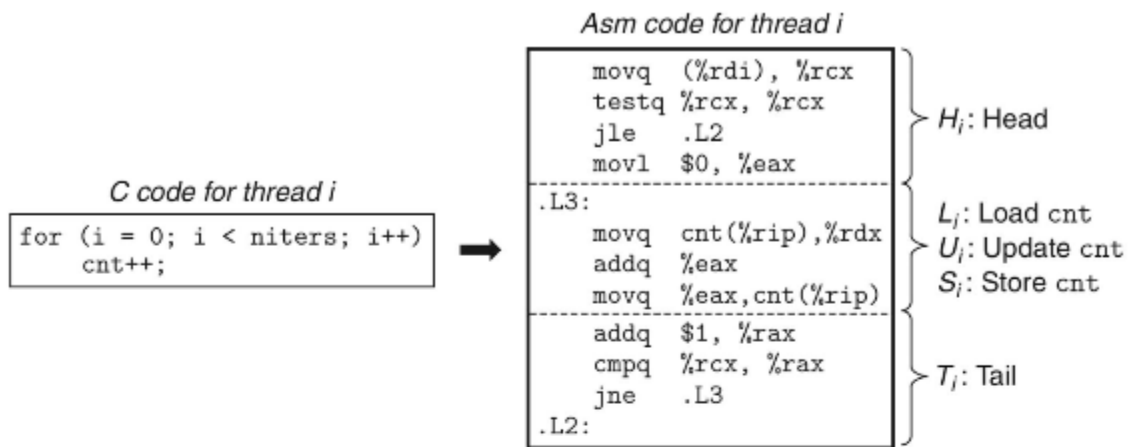


Figure 12.17 Assembly code for the counter loop (lines 40–41) in `badcnt.c`.

Step	Thread	Instr.	%rdx ₁	%rdx ₂	cnt
1	1	H_1	-----	-----	0
2	1	L_1			
3	2	H_2			
4	2	L_2			
5	2	U_2			
6	2	S_2			
7	1	U_1			
8	1	S_1			
9	1	T_1			
10	1	T_2			

What is the value of `cnt` at the end of this ordering? Is the result surprising?