CS 4400: Computer Systems

Finall Exam SAMPLE SOLUTIONS
Fall 2010

1. (a) VPN: [15-9], VPO: [8-0], TLBT: [15-10], TLBI: [9]

   (b) PPN: [12-9], PPO: [8-0], CT: [12-5], CI: [4-2], CO: [1-0]

   (c) 0011 0001 1101 1110

   (d)

Parameter	Value
VPN	0x18
TLB Index	0x00
TLB Tag	0x0C
TLB Hit? (Y/N)	N
Page Fault? (Y/N)	N
PPN	0x1

   (e) 0 0011 1101 1110

   (f)

Parameter	Value
Byte offset	0x2
Cache Index	0x7
Cache Tag	0x1E
Cache Hit? (Y/N)	Y
Cache Byte returned	0x0F

2.

address	*contents*
0x400b030	allocated block's footer, changed to: 0x15
0x400b02c	allocated block's payload/padding, must be same as before
0x400b028	allocated block's payload/padding, must be same as before
0x400b024	allocated block's header, changed to: 0x15
0x400b020	new free block's footer: 0x2e
0x400b01c	garbage, leave same value as before
0x400b018	garbage, leave same value as before
0x400b014	garbage, leave same value as before
0x400b010	garbage, leave same value as before
0x400b00c	garbage, leave same value as before
0x400b008	garbage, leave same value as before
0x400b004	garbage, leave same value as before
0x400b000	garbage, leave same value as before
0x400affc	new free block's header: 0x2e

3. For `size` function, change to `return (*(int*)hp) & (~0x7);`. Function `footer` is correct. For `prev_allocated` function, change to `return ((*(int*)hp) & 0x2) >> 1;`.

4. (a) i. Thread contexts are much smaller than process contexts. This is primarily due the to fact that all threads running in a process share the entire virtual memory address space.

ii. Sharing data between threads is easier and incurs less overhead than between processes because threads share the same virtual memory address space.

(b) i. A lock is not required for correctness if all threads accessing the shared variable do only reads.

ii. The semaphore is itself a shared variable. Without atomicity, it could suffer the same interruptions in the load/update/store sequence that we must prevent for other shared variables to ensure correctness.

(c) Server B will perform better than server A is there is a steady stream of eight or fewer concurrent requests, because B does not incur the overhead of thread creation and removal for each request (as A does).

Server A will perform better than server B if there are regularly more than eight concurrent requests, because A can dynamically create a new thread to handle each new request.

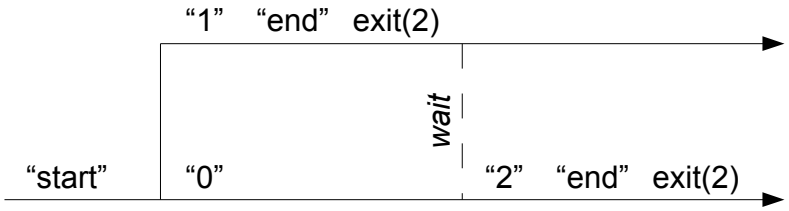
5. might, might, never

6. Assembler routine `foo1` corresponds to C function `choice3`. Assembler routine `foo2` corresponds to C function `choice5`. Assembler routine `foo3` corresponds to C function `choice1`.

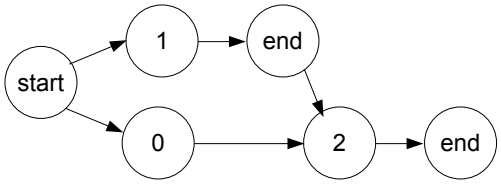
7. `int a`, `short b` (or `char b`), `double d`

8. (a) `buf[0] = 0x64636261`, `buf[1] = 0x68676665`, `buf[2] = 0x08040069`
   (b) `%ebp = 0x68676665`
   (c) `%eip = 0x08040069`

9. (a)



(b) All three valid topological sorts of the following graph.



start	start	start
1	0	1
end	1	0
0	end	end
2	2	2
end	end	end