Assignment #4

Tianhao Liu

[liux3271@umn.edu](mailto:liux3271@umn.edu)

Section 014 (02:30 PM on Thursday)

Problem #1

A. Virtual address

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| TLBT | | | | | | TLBI | VPO | | | | | | | | |
| VPN | | | | | | | VPO | | | | | | | | |

B. Physical address

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| PPN | | | | PPO | | | | | | | | |
| CT | | | | | | | | CI | | | CO | |

C. 0x3925

Binary: 001110 0 100100101

VPN: 0x1c

TLBI: 0x0, TLBT: 0xe -> TLB MISS

Page fault

D. 0x1330

Binary: 000100 1 100110000

VPN: 0x9

TLBI: 0x1, TLBT: 0x4 -> TLB MISS

No Page fault

PPN: 0x8

Physical address: 1000 100110000

CT: 0x89, CI:0x4, CO: 0x0 -> Cache Miss

E. 0x1383

Binary: 000100 1 110000011

VPN: 0x9

TLBI: 0x1, TLBT: 0x4 -> TLB MISS

No Page fault

PPN: 0x8

Physical address: 1000 110000011

CT: 0x8c, CI:0x0, CO: 0x3 -> Cache Miss

F. 0x0766

Binary: 000001 1 101100110

VPN: 0x03

TLBI: 0x1, TLBT: 0x1 -> TLB HIT

No Page fault

PPN: 0x5

Physical address: 0101 101100110

CT: 0x5b, CI:0x1, CO: 0x2 -> Cache Miss

Problem #2

In alloc():

ret = root;

root = root->next\_free;

In dealloc():

p->next\_free = root;

root = p;

Problem #4

// When using integer data, date\_t = int

// When using double data, date\_t = double

void inner (vec\_ptr u, vec\_ptr v, data\_t \*dest) {

long int i;

int length = vec\_length(u);

int len6 = length - (length % 6);

data\_t \*udata = get\_vec\_start(u);

data\_t \*vdata = get\_vec\_start(v);

data\_t sum1 = (data\_t) 0;

data\_t sum2 = (data\_t) 0;

data\_t sum3 = (data\_t) 0;

data\_t sum4 = (data\_t) 0;

data\_t sum5 = (data\_t) 0;

data\_t sum6 = (data\_t) 0;

for (i = 0; i < len6; i += 6){

sum1 = sum1 + udata[i] \* vdata[i];

sum2 = sum2 + udata[i+1] \* vdata[i+1];

sum3 = sum3 + udata[i+2] \* vdata[i+2];

sum4 = sum4 + udata[i+3] \* vdata[i+3];

sum5 = sum5 + udata[i+4] \* vdata[i+4];

sum6 = sum6 + udata[i+5] \* vdata[i+5];

}

for (; i < length; i++){

sum1 = sum1 + udata[i] \* vdata[i];

}

\*dest = (sum1 + sum2) + (sum3 + sum4) + (sum5 + sum6);

}