

Name: _____

Directions: **Work only on this sheet** (on both sides, if needed). MAKE SURE TO COPY YOUR ANSWERS TO A SEPARATE SHEET FOR SENDING ME AN ELECTRONIC COPY LATER.

On all Tests, 32-bit word size on Intel machines running Linux is assumed unless otherwise stated.

1. (15) Say we have a multicore machine. Among the registers EAX, EBP, EFLAGS, ESP, ESI, IDT and PTR, state which must have separate versions in each core. For example, if you state EAX has such a property, that means that each core must have its own separate EAX register, rather than one EAX register serving all cores. Hint: There will be at least one register in the list with this property.

2. (20) How many times will the hardware consult the page table (assume no TLB or other cache) during the execution (not decode) of each of the following instructions:

- (a) ()
 incl %eax
- (b) ()
 addl (%eax), %ebx
- (c) ()
 movl x, %ebx
- (d) ()
 movl \$x, %ebx

3. Consider the code

```
int colsum(int *x, int nrow, int ncol,
           int whichcol)
{
    int i, sum = 0;
    for (i = 0; i < nrow; i++) {
        sum += x[nrow*i + whichcol];
    }
    return sum;
}
```

Running this through **gcc -S** yields:¹

```
colsum :
    pushl   %ebp
    movl   %esp, %ebp
    subl   $16, %esp
    movl   $0, -8(%ebp)
    movl   $0, -4(%ebp)
    jmp    .L2
.L3:
    movl   12(%ebp), %eax
    imull  blank(a)(%ebp), %eax
    addl   20(%ebp), %eax
```

```
    sll    $blank(b), %eax
    addl   8(%ebp), %eax
    movl   (%eax), %eax
    addl   %eax, -8(%ebp)
    addl   blank(c), -4(%ebp)
.L2:
    movl   -4(%ebp), %eax
    cmpl   12(%ebp), %eax
    jl     blank(d)
    movl   -8(%ebp), %eax
    leave
    ret
```

- (a) (15) Fill blank (a).
(b) (15) Fill blank (b).
(c) (15) Fill blank (c).
(d) (15) Fill blank (d).
(e) (5) Fill blank (e).

Solutions:

1. all but IDT

2. 0, 1, 1, 0

3.

```
colsum :
    pushl   %ebp
    movl   %esp, %ebp
    subl   $16, %esp
    movl   $0, -8(%ebp)
    movl   $0, -4(%ebp)
    jmp    .L2
.L3:
    movl   12(%ebp), %eax
    imull  -4(%ebp), %eax
    addl   20(%ebp), %eax
    sll    $2, %eax
    addl   8(%ebp), %eax
    movl   (%eax), %eax
    addl   %eax, -8(%ebp)
    addl   $1, -4(%ebp)
.L2:
    movl   -4(%ebp), %eax
    cmpl   12(%ebp), %eax
    jl     .L3
    movl   -8(%ebp), %eax
    leave
    ret
```

¹I've removed some extraneous material.