

**Hacettepe Üniversitesi  
Bilgisayar Mühendisliği Bölümü  
BBM341 Sistem Programlama  
Bütünleme Sınavı – 31 Ocak 2014**

```
leal (%eax,%eax,2), %eax
sall $2, %eax
```

```
t <- x+x*2
return t << 2;
```

**Soru 1.** Yukarıdaki örnekte %eax yazmacı  $12_{10}$  ile çarpılmaktadır. Benzer biçimde %eax yazmacını  $6_{10}$  ile çarpan kodu yazınız.

**Soru 2.** Aşağıda onlu tabanda verilen sayıların ikili tabanda karşılıklarını yazınız.

[illegible]

<b>Normalized Values</b> <b>Condition: <math>\text{exp} \neq 000 \dots 0</math></b> <b>and <math>\text{exp} \neq 111 \dots 1</math></b> <b><math>E = \text{exp} - \text{Bias}</math></b>	<b>Denormalized Values</b> <b>Condition:</b> <b><math>\text{exp} = 000 \dots 0</math></b> <b><math>E = -\text{Bias} + 1</math></b>	<b>Bias = <math>2^{k-1} - 1</math></b>	<p>The diagram shows a horizontal bar divided into three sections: a yellow section labeled 's' (sign) with width '1', a red section labeled 'exp' (exponent) with width '4-bits', and a blue section labeled 'frac' (fraction) with width '3-bits'.</p>
---	---	--	--

**Soru 3.** Yukarıdaki bilgiler kapsamında  $1_{10}$  sayısını 8 bitlik kayan noktalı sayı olarak kodlayınız.

**Soru 4.** Aşağıdaki örnekte iki sayının arasındaki farkı bulan fonksiyonun C programlama dili ve x86 simgesel dili ile kodlamaları verilmiştir. Siz de üç sayının en büyüğünü bulan fonksiyon için kodlamaları C programlama dili ve x86 simgesel dili ile yapınız.

```
int absdiff(int x, int y)
{
    int result;
    if (x > y) {
        result = x-y;
    } else {
        result = y-x;
    }
    return result;
}
```

```
absdiff:
    pushl   %ebp
    movl    %esp, %ebp
    movl    8(%ebp), %edx
    movl    12(%ebp), %eax
    cmpl    %eax, %edx
    jle     .L6
    subl    %eax, %edx
    movl    %edx, %eax
    jmp     .L7
.L6:
    subl    %edx, %eax
.L7:
    popl    %ebp
    ret
```

**Soru 5.** Soru 4’de verilen örnek için yığıt yapısını çiziniz.

**Soru 6.** Soru 4’deki çözümünüz için yığıt yapısını çiziniz.

**Soru 7.** Soru 4'deki çözümünüzde geliştirdiğiniz kodu çağırın kesimi x86 simgesel dili ile kodlayınız.

**Soru 8.** Aşağıdaki kod kesimi uygulandığında olası bir çıktısını veriniz.

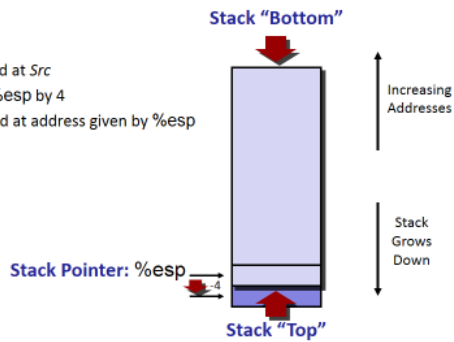
```
void fork4()
{
    printf("L0\n");
    if (fork() != 0) {
        printf("L1\n");
        if (fork() != 0) {
            printf("L2\n");
            fork();
        }
    }
    printf("Bye\n");
}
```

**Soru 9.** Görev anahtarlama (*Context Switching*) sırasında uygulanan adımları listeleyiniz.

## IA32 Stack: Push

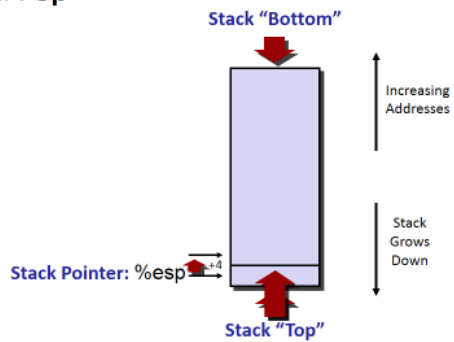
### ■ `pushl Src`

- Fetch operand at *Src*
- Decrement `%esp` by 4
- Write operand at address given by `%esp`



43

## IA32 Stack: Pop



45

## Procedure Control Flow

- Use stack to support procedure call and return

### ■ Procedure call: `call label`

- Push return address on stack
- Jump to *label*

### ■ Return address:

- Address of the next instruction right after call
- Example from disassembly

```
804854e: e8 3d 06 00 00  call  8048b90 <main>
8048553: 50
```

- Return address = `0x8048553`

### ■ Procedure return: `ret`

- Pop address from stack
- Jump to address

47