

SAKARYA ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ 2019-2020 YAZ OKULU
SİSTEM PROGRAMLAMA FİNAL SINAVI

Süre: 70 dk. Başarılar Dilerim.

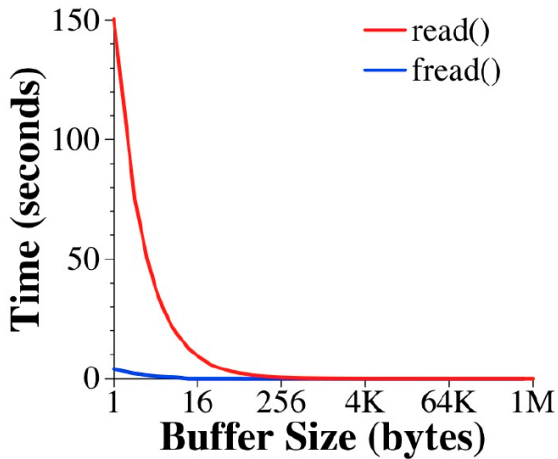
Not: Cevap kâğıtlarınızın net okunabilir bir şekilde resmini çekerek, ogrencino.zip isimindeki tekil dosyayı sınav süresinin bitimine kadar sisteme yüklemiş olmanız gerekmektedir. Bu süre zarfında sisteme yüklenmeyen sınav kâğıtları değerlendirilmeyecektir.

1- (15 puan)

```
#include <stdio.h>
#include <stdlib.h>
#include "fields.h"
int main(int argc, char **argv)
{
    IS is;
    int i, n;
    char *lines[10];
    is = new_inputstruct(NULL);
    n = 0;
    while (get_line(is) >= 0) {
        lines[n%10] = is->text1;
        n++;
    }
    i = (n >= 10) ? (n-10) : 0;
    for ( ; i < n; i++) printf("%s", lines[i%10]);
    exit(0);
}
```

Yukarıdaki programda libfdt kütüphanesi kullanılarak bir veri dosyasından satırlar okunup, dosyada bulunan son on satır ekrana bastırılmak isteniyor. Fakat bu kod parçasındaki bir eksiklikten dolayı istenilen çıktı elde edilememektedir. Buradaki problemi ve nasıl düzeltilebileceğini açıklayınız. Cevap yukarıdaki alana yazılmalıdır.

2- (20 puan)



a-) Yukarıdaki grafiği kısaca yorumlayınız. (7 puan)

b-) Sistem çağrılarının neden maliyetli olduğunu açıklayınız. (7 puan)

c-) Buffer kullanımının önemini kısaca anlatınız. (6 puan)

- 3- (25 puan) C programlama dilinde program çalıştırılırken aldığı parametre değerinden 1'er geriye doğru sayan bir program yazınız. Bu program çalışırken Ctrl-C tuşuna basıldığında sinyale ait icrayı değiştirerek int döngü değişkeninin o anki değerini 2 azaltılacak ve program kaldığı yerden yeni değeriyle 1'er geriye doğru saymaya devam edecektir.

Örnek çalıştırma: user\$>./p1 50

0	1	2	3	4
<pre>int fonk(int i, int j) { int k; k = (i-2)*(j+5)+11; return k; } main(){ int y; y=fonk(1,5); }</pre>	<pre>int fonk(int i, int j) { int k; k = (i-2)*j+5*i; return k; } main(){ int y; y=fonk(2,4); }</pre>	<pre>int fonk(int i, int j) { int k; k = 2*i+5*j-5; return k; } main(){ int y; y=fonk(3,7); }</pre>	<pre>int fonk(int i, int j) { int k; k = 2*i-5*(j+5); return k; } main(){ int y; y=fonk(4,3); }</pre>	<pre>int fonk(int i, int j) { int k; k = 2*i+5*(j+3); return k; } main(){ int y; y=fonk(7,5); }</pre>
5	6	7	8	9
<pre>int fonk(int i, int j) { int k; k = (i+5)*j+2*j; return k; } main(){ int y; y=fonk(8,5); }</pre>	<pre>int fonk(int i, int j) { int k; k = (i+j)*7+5*i; return k; } main(){ int y; y=fonk(6,4); }</pre>	<pre>int fonk(int i, int j) { int k; k = 2+i*(5+j)+5; return k; } main(){ int y; y=fonk(9,5); }</pre>	<pre>int fonk(int i, int j) { int k; k = 2*i-5*j+5; return k; } main(){ int y; y=fonk(7,8); }</pre>	<pre>int fonk(int i, int j) { int k; k = 5*(i+j)+12*j; return k; } main(){ int y; y=fonk(1,4); }</pre>

4- (40 puan) Yukarıda verilmiş olan C dilinde yazılmış programlardan öğrenci numaranız ile elde edeceğiniz değere ait assembler kodunu yazınız. (örnek: b12001738 %10 = 8 nolu kodun karşılığı yazılacak)

- Ders kapsamında gösterilen mimariye uygun kod yazımı gerçekleştirilmelidir.
- Öğrenci numaranızı kullanarak elde ettiğiniz şıkkın dışında yapılacak çözümler değerlendirilmeyecek.
- Çözüm için yazılan assembler kodda dersteki yapılan örneklere benzer şekilde her bir C kod satırının işlemlerine karşılık gelecek assembler karşılıkları gösterilmelidir.
- Bu program çalıştırıldığında, bellek üzerindeki işlemler aşağıdaki verilen şablon stack üzerinde adım adım gösterilmelidir. Aşağıda verilen şablon stack sayısını adım sayısına göre artırınız.
(Puanlama-> asm kodunun yazılması (15 puan), c kodundaki her bir satıra karşılık gelen asm karşılığının gösterilmesi(10 puan), stack üzerindeki işlemlerin adım adım gösterilmesi (15 puan))

