

## Soru 1

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <errno.h>
5  #include <sys/stat.h>
6
7  main()
8  -{
9      char mode[]="0744";
10     char buf[100]="/home/sau/test.txt";
11     int i;
12
13     i = atoi(mode);
14     if (chmod (buf,i) < 0)
15         printf("Hata chmod %d", errno);
16 }
```

Şekildeki C programı ile "test.txt" adlı dosyanın erişim modu değiştirilmek isteniyor. Hatalı satır numarası ve hata aşağıdakilerden hangisi olabilir?

- A ☐ Satır 13: atoi() fonksiyonu mod bitlerinin dönüşümü için uygun değildir.
- B ☐ Satır 10: Dosya adının tutulduğu buffer uygun değildir.
- C ☐ Satır 5: sys/stat.h yerine sys/mode.h kütüphanesi eklenmeliydi.
- D ☐ Satır 15: "errno" değişkeni tanımlanmadan kullanılmış.
- E ☐ Satır 14: chmod() eksik parametre ile kullanılmış.

## Soru 2

Bir Linux sisteminde **umask** deęiřkeni řu anda 077 deęerine ayarlanmıř olsun. "**sau**" adlı bir kullanıcı (**sau** aynı zamanda "**users**" grubunun bir üyesidir) "**veri.txt**" adlı bir dosya oluřturduęunda, "**users**" grubunun üyesi olan kullanıcılar bu dosyayla ne yapabilirler?

- A** ☐ Dosyaya hiębir řekilde eriřimleri yoktur.
- B** ☐ Dosyayı görüntüleyebilirler ve yürütebilirler ancak deęiřtiremez veya kaydedemezler.
- C** ☐ Dosyayı görüntüleyebilirler, ancak deęiřtiremez veya kaydedemezler.
- D** ☐ Dosyayı açabilir, deęiřtirebilir ve kaydedebilirler.
- E** ☐ Yürütülebilir bir dosyaysa, dosyayı da yürütebilirler.

Seęimi Boř Bırakmak İstiyorum

### Soru 3

Bir dizinin içindeki dosya ve dizin adlarına erişilmek istenirse hangi veri yapısına ihtiyaç vardır?

- A** ☐ dirent
- B** ☐ lstat
- C** ☐ inode
- D** ☐ stat
- E** ☐ opendir

Seçimi Boş Bırakmak İstiyorum

#### Soru 4

```
#include<stdio.h>

void foo()
{
    char *ch = 0;
    *ch = 3;
}

int main()
{
    foo();
    printf("Tamamdır\n");
    return 0;
}
```

Şekildeki resimde verilen program derlenip çalıştırılmak isteniyor. A) Derleme, B) Çalışma işlemleri ve ekran çıktıları hakkında aşağıdakilerden hangisi doğrudur?

- A** ☐ A: Program sorunsuz derlenebilir, B: Program "Bus error" hatası verir.
- B** ☐ A: Program sorunsuz derlenebilir, B: Program "Segmentation fault" hatası verir.
- C** ☐ A: Program sorunsuz derlenebilir, B: Program foo() fonksiyonunda takılır kalır.
- D** ☐ A: Program derleme hatası verir, B: Yürütülebilir dosya olmadığından program çalışmaz.
- E** ☐ A: Program sorunsuz derlenebilir, B: Program çalışır ve ekrana "Tamamdır" yazılır.

Seçimi Boş Bırakmak İstiyorum

### Soru 5

```
int foo(int a,int b,int c)
{
    int d;
    d=(a+b)/c;
    return d;
}
int p;
int main() {

    int x;
    x=foo(5,6,7);
    return 0;
}
```

Şekilde verilen programın **foo()** fonksiyonunu işletirken **b** lokal değişkeninin adresi simgesel dilde (assembler) aşağıdakilerden hangisidir?

- A ☐ fp+12
- B ☐ fp-8
- C ☐ fp
- D ☐ fp+16
- E ☐ fp+8

Seçimi Boş Bırakmak İstiyorum



## Soru 6

```
1  #include<stdio.h>
2  #include<string.h>
3
4  int main()
5  -{
6      char kaynak[] = "Sistem Programlama";
7      char *hedef = strdup(kaynak);
8      printf("%s", hedef);
9      return 0;
10 }
```

Şekilde verilen program derlenip çalıştırıldığında ekranda ne görünür?

- A ☐ amalmargorP metsiS
- B ☐ Sistem
- C ☐ Programlama
- D ☐ Sistem Programlama
- E ☐ Sistem  
Programlama

Seçimi Boş Bırakmak İstiyorum

## Soru 7

```
int foo(int a,int b,int c)
{
    int d;
    d=(a+b)/c;
    return d;
}
int p;
int main() {

    int x;
    x=foo(5,6,7);
    return 0;
}
```

Şekildeki program işletilirken p değişkenine ait adres aşağıdakilerden hangisidir?

- A ☐ fp-4
- B ☐ fp-8
- C ☐ Hiçbiri
- D ☐ fp
- E ☐ fp+4

Seçimi Boş Bırakmak İstiyorum

## Soru 8

Puan: 10,00

Linux sisteminde, bulunduğumuz dizinde yer alan "fl.txt" adlı dosyanın hard-link sayısını bulup ekrana yazdıran programı C dilinde yazınız?

(**Not:** Programda "system()" çağrısı kullanılarak yapılan çözümler kabul edilmeyecektir)

Kaydet

Her değişiklik yaptığınızda kaydet butonuna basmalısınız.



## Soru 9

```
struct dirent {
    ino_t      d_ino;      /* inode number */
    off_t      d_off;      /* offset to the next dirent */
    unsigned short d_reclen; /* length of this record */
    unsigned char d_type;   /* type of file; not supported
                           by all file system types */
    char        d_name[256]; /* filename */
};
```

Linux'te var olan ve şekilde görülen dirent veri yapısı bir çalışma ile değiştirilse ve buraya inode no'su yerine, tamamen inode structure'i konsa ve bu şekilde yeni bir dosya sistemi oluşturulsa, bunun avantaj ve dezavantajları ne olurdu?

- A** ☐ Avantajları: Dosya içeriklerine ulaşırken disk üzerinde arama sayısı azalabilirdi.  
Dezavantaj: Hard link oluşturmak imkansızlaşırđı. Inode'ları bulmak zorlaşırđı.
- B** ☐ Avantajları: Dosya içerikleri azalabilirdi.  
Dezavantaj: Inode'ları bulmak zorlaşırđı.
- C** ☐ Avantajları: Hiç bir avantajı olmazdı.  
Dezavantaj: Dizin dosyalarının boyutları büyürdü.
- D** ☐ Avantajları: Dosya içeriklerine ulaşırken disk üzerinde arama sayısı azalabilirdi.  
Dezavantaj: Dezavantajı yoktur.
- E** ☐ Avantajları: Dosya değışim bilgilerine daha kolay ulaşılabilirdi.  
Dezavantaj: Soft link oluşturmak zorlaşırđı.

Seçimi Boş Bırakmak İstiyorum

## Soru 10

```
int main() // Bu programın yürütülebilir dosya adı "p1"
{
    int i; FILE *f;
    f = fopen("/home/sau/veri.txt", "r");
    if (f == NULL) {
        printf("f = null.  errno = %d\n", errno);
        perror("hata");
    }
    return 0;
}
```

U: --- p1.c Top L7 (C/\*l Abbrev)

\$ > echo "" > /home/sau/veri.txt

\$ > chmod 0 /home/sau/veri.txt

\$ > ./p1

1

\$ > rm -f /home/sau/veri.txt

\$ > ./p1

2

Şekilde verilen program derlendiğinde "p1" adında yürütülebilir program elde edilmektedir. Buna göre Linux komut satırında görülen işlemler yapılırsa, şekilde gösterilen 1 ve 2 nolu yerlerde ne tür çıktılar alınır?

- A** ☐ 1: hata: Bad file number  
2: hata: Operation not permitted
- B** ☐ 1: Error: Permission denied  
2: Error: No such file or directory
- C** ☐ 1: hata: Operation not permitted,  
2: hata: Permission denied
- D** ☐ 1: hata: Permission denied  
2: hata: No such file or directory
- E** ☐ 1: hata: Bad file number,  
2: hata: No such file or directory

## Soru 11

```
1 void a(int k) {
2     int i = 0;
3     if(k > 1)
4         i = k + k;
5     return i;
6 }
7 void main() {
8     int i, j;
9     j = 2;
10    i = a(j);
11 }
```

```
1 a:
2     push #4
3     st %g0 -> [fp]
4     _____
5     _____
6     _____
7     ld [fp+12] -> %r0
8     add %r0, %r0 -> %r0
9 ll:
10    ret
11 main:
12    push #8
13    mov #2 -> %r0
14    st %r0 -> [fp]
15    ld [fp] -> %r0
16    st %r0 -> [sp]--
17    jsr a
18    pop #4
19    st %r0 -> [fp-4]
20    ret
```

Şekilde verilen C programının karşılığı olan simgesel dil (assembler) programda, boş bırakılan satırların yerine aşağıdakilerden hangi satırlar gelmelidir?

- A** ☐ ld [fp+12] -> %r0  
mul %r0, %r0 -> %r0  
ble ll
- B** ☐ ld [fp+12] -> %r0  
add %r0, %g1 -> %r0  
ble ll
- C** ☐ cmp %r0, %g1  
ble ll  
ld [fp+12] -> %r0
- D** ☐ ld [fp+12] -> %r1  
cmp %r0, %g1  
ble ll
- E** ☐ ld [fp+12] -> %r0  
cmp %r0, %g1  
ble ll

## Soru 12

```
TARGET=pro
```

```
all: $(TARGET)
```

```
$(TARGET): $(TARGET).c
```

---

```
clean:
```

```
rm -rf $(TARGET)
```

Şekilde verilen Makefile’da düz çizgi ile verilen boş satıra;

A: hangisi gelebilir,

B: Derleme sonucunda yürütülebilir program adı ne olur?

- A** ☐ A: gcc pl.c -o \$@  
B: pro
- B** ☐ A: gcc prol.c -o \$@  
B: pl
- C** ☐ A: gcc all.c -o \$@  
B: pro
- D** ☐ A: gcc \$(TARGET).c -o \$@  
B: pl
- E** ☐ A: gcc \$(TARGET).c -o \$@  
B: pro



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>

Şekilde verilmiş olan C programlardan öğrenci numaranız ile elde edeceğiniz koda ait assembler kodunu yazınız. (örnek: b19001736 %10 = 6 nolu kodun karşılığı yazılacak)

- Ders kapsamında gösterilen mimariye uygun kod yazımı gerçekleştirilmelidir.
- Öğrenci numaranızı kullanarak elde ettiğiniz kodun çözümünü dışında yapılacak çözümler değerlendirilmeyecek.
- Çözüm için yazılan assembler kodda dersteki yapılan örneklerle benzer şekilde her bir C kod satırının işlemlerine karşılık gelecek assembler karşılıkları gösterilmelidir. (aksi takdirde puan kılacaktır.)

#### Soru 14

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 int main()
5 {
6     int i;
7     char j[14];
8     int *ip;
9     char *jp;
10    ...
11    return 0;
12 }
```

64 bitlik bir sistem için hazırlanan ve şekilde bir kısmı gösterilen programda yapılan tanımlamalara göre aşağıdakilerden hangisi **illegal** bir atamadır?

- A ☐ `ip = 0x7fff57350268;`
- B ☐ `jp = 0x7fff5735025;`
- C ☐ `jp = j;`
- D ☐ `ip = &i;`
- E ☐ `jp = &j;`

Seçimi Boş Bırakmak İstiyorum



### Soru 15

```
#include<stdio.h>
#include<stdlib.h>
A
_____

void foo(int a)
{
    printf("Çıkıyoruz\n");
    exit(1);
}

int main()
{
B
_____
    while(1) ;
    return 0;
}
```

Şekildeki program "CTRL-C" ile klavyeden gelecek kesmelere duyarlı yapılmak isteniyor. Buna göre eksik satırlara aşağıdakilerden hangisi gelmelidir?

- A** ☐ A: #include < unistd.h >  
B: signal(SIGINT, foo);
- B** ☐ A: #include < sig.h >  
B: signal(SIGINT, foo);
- C** ☐ A: #include < unistd.h >  
B: signal(SIGALRM, foo);
- D** ☐ A: #include < signal.h >  
B: signal(SIGINT, foo);
- E** ☐ A: #include < signal.h >  
B: signal(SIGTERM, foo);

### Soru 16

```
int foo(int a,int b,int c)
{
    int d;
    d=(a+b)/c;
    return d;
}
int p;
int main() {

    int x;
    x=foo(5,6,7);
    return 0;
}
```

Şekilde verilen programda `foo()` fonksiyonunu işletirken `d` lokal değişkeninin adresi simgesel dilde (assembler) aşağıdakilerden hangisidir?

- A ☐ `fp+8`
- B ☐ `fp`
- C ☐ `fp-8`
- D ☐ `fp+4`
- E ☐ `fp-4`

Seçimi Boş Bırakmak İstiyorum

## Soru 17

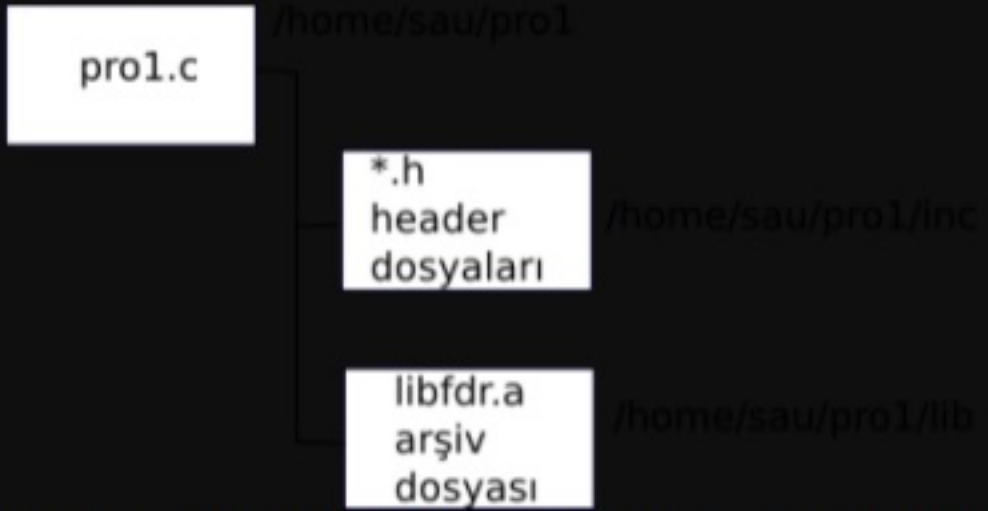
Bellek sızıntısı (memory leak) kavramı aşağıdaki şıklardan hangisinde doğru tanımlanmıştır?

- A** ☐ Bellek segment ihlali oluşması.
- B** ☐ Bellekte yığıt (heap) veya yığın (stack) üzerinde yanlış bir adrese erişilmeye çalışılması.
- C** ☐ Yanlış bellek tahsis işlemi yapılması.
- D** ☐ Bellekte yığıt (heap) bölgesinde kullanılmayan verinin atıl bir şekilde kalması.
- E** ☐ Bellekte yığıt (heap) bölgesine fazla veri konulmaya çalışılması.

Seçimi Boş Bırakmak İstiyorum

**Soru 18**

Puan: 2,00



Dosya erişimi ile ilgili **pro1.c** adında bir program yazılmış, bu programda **libfdr** fonksiyonları kullanılmıştır. **pro1.c** adlı program dosyasının, header dosyalarının (**\*.h** dosyaları) ve arşiv dosyasının (**libfdr.a** dosyası) konumları şekilde verilmiştir. Derleme ve link sonrası yürütülebilir program (**pro1**) **pro1.c** ile aynı dizinde olacaktır. Buna göre **pro1.c** adlı C programının **derleme satırı** aşağıdakilerden hangisi olabilir?

- A** ☐ \$ > gcc pro1.c -lfdr -o pro1
- B** ☐ \$ > gcc -L./inc pro1.c -I./lib -lfdr -o pro1
- C** ☐ \$ > gcc -I./inc pro1.c -L./lib -libfdr -o pro1
- D** ☐ \$ > gcc pro1.c -o pro1 -libfdr
- E** ☐ \$ > gcc -I./inc pro1.c -L./lib -lfdr -o pro1

[Seçimi Boş Bırakmak İstiyorum](#)

### Soru 19

```
#include <stdio.h>

int main()
{
    int a = 4;
    double b;
    int *ptr1;
    double *ptr2;

    ptr1 = &a;
    ptr2 = &b;

    printf("ptr1 önce 0x%x, ", ptr1);
    ptr1++;
    printf("sonra: 0x%x\n", ptr1);

    printf("ptr2 önce: 0x%x, ", ptr2);
    ptr2++;
    printf("sonra 0x%x\n", ptr2);

    return 0;
}
```

Program çıktısı:

```
ozmen@DESKTOP-J11TOIF:~/sisProg2021$ ./p1
ptr1 önce 0xd4609c0c, sonra: A
ptr2 önce: 0xd4609c10, sonra B
```

Şekildeki program 64-bit bir Linux işletim sisteminde gcc ile derlenip çalıştırıldığında aynı şekilde aşağıda gösterildiği gibi bir çıktı üretmektedir. Buna göre boş bırakılan A, ve B ile etiketlenmiş yerlerde ne görülür?

- A** ☐ A: 0xd4609c0e B: 0xd4609c18
- B** ☐ A: 0xd4609c0d B: 0xd4609c11
- C** ☐ A: 0xd4609c0f B: 0xd4609c18
- D** ☐ A: 0xd4609c10 B: 0xd4609c18
- E** ☐ A: 0xd4609c10 B: 0xd4609c14

Seçimi Boş Bırakmak İstiyorum



```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <dirent.h>
4  #include <sys/stat.h>
5  #include <string.h>
6
7  - long get_size(const char *fn) {
8      DIR *d;
9      struct dirent *de;
10     struct stat buf;
11     int exists;
12     long total_size;
13     int fn_size;
14     char *dir_fn;
15     int dir_fn_size;
16     int sz;
17     d = opendir(fn);
18     total_size = 0;
19     fn_size = strlen(fn);
20     dir_fn_size = fn_size + 10;
21     dir_fn = (char *) malloc(sizeof(char) * dir_fn_size);
22     strcpy(dir_fn, fn);
23     strcat(dir_fn + fn_size, "/");
24     - for (de = readdir(d); de != NULL; de = readdir(d)) {
25         sz = strlen(de->d_name);
26         - if (dir_fn_size < fn_size + sz + 2) {
27             dir_fn_size = fn_size + sz + 10;
28             dir_fn = realloc(dir_fn, dir_fn_size);
29         }
30         strcpy(dir_fn + fn_size + 1, de->d_name);
31         exists = stat(dir_fn, &buf);
32         total_size += buf.st_size;
33         - if (S_ISDIR(buf.st_mode) && strcmp(de->d_name, ".") != 0 && strcmp(de->d_name, "..") != 0) {
34             total_size += get_size(dir_fn);
35         }
36     }
37     closedir(d);
38     free(dir_fn);
39     return total_size;
40 }
41
42 - int main() {
43     long total_size;
44     total_size = get_size(".");
45     printf("%ld\n", total_size);
46     return 0;
47 }

```

Şekildeki program bir dizin içindeki dosyaların ve dizin-dosyalarının disk üzerinde kapladığı toplam alanı bulmak için geliştirilmiştir. Ancak çok fazla dizin iç-içe olunca "Çok fazla açık dosya" hatası vererek yürütmeyi durdurmaktadır. Buna göre ne yapılmalıdır?

- A** ☐ 37. satırdaki closedir() uygun düzenleme ile 39. satırdaki return total\_size çağrısının önüne taşınmalı.
- B** ☐ 37. satırdaki closedir() uygun düzenleme ile 33. satırdaki get\_size() rekürsive çağrısının önüne taşınmalı.
- C** ☐ 33. satırdaki "if" ifadesinin koşulu uygun şekilde ayarlanarak, dizin dosyaları tekrar açılmamalıdır.
- D** ☐ 23. satırda dizin adları oluşturulurken, iç dizinler doğru oluşturulamadığından tekrarlı dosya açmaları olmaktadır. Burada düzenleme yapılmalıdır.
- E** ☐ 31. satırda yer alan stat() çağrısı yerine lstat() kullanılmalıdır.



## Soru 21

`int fd = open("f1.txt", O_WRONLY | O_CREAT, 0640);` sistem çağrısı için aşağıdakilerden hangisi doğrudur?

- A** ☐ "f1.txt" dosyası kendime okuma/yazma, gruba ve herkese sadece okuma modunda açılır boş olarak açılır.
- B** ☐ "f1.txt" dosyası okumak/yazmak için açılır, dosyada veri varsa sıfırlanarak (boş olarak) açılır.
- C** ☐ "f1.txt" dosyası kesilerek yazmak için açılır, yoksa hata döner.
- D** ☐ "f1.txt" adında dosya yoksa oluşturulur, varsa sıfırlanarak (boş olarak) ve yazmak için açılır.
- E** ☐ "f1.txt" adlı dosya varsa yazmak için açılır, yoksa önce oluşturulur ve yazmak için açılır.

Seçimi Boş Bırakmak İstiyorum



## Soru 23

Puan: 15,00

```
number.txt - Not Defteri
Dosya  Düzen  Biçim  Görünüm  Yardım
63  83  17  83  71  81
73  101  18  8  98
34  28  29  84  291
. . . .
```

Bu soruda derste anlatılan "fields" ve "dlist" kütüphanelerini kullanarak bir program yazmanız istenilmektedir, programın adı **p1.c**, yürütülebilir dosyanın adı da **p1** olsun. Program çalıştırılırken verilerin bulunduğu dosya kullanıcı tarafından parametre olarak verilecektir. Örneğin:

```
$ > p1 veri.txt
```

Veri dosyasının içeriği şekilde verildiği gibidir, her satırında boşlukla ayrılmış tamsayılar bulunmaktadır. Programınız veri dosyasında bulunan sayıları satır satır okuyarak, her bir satırdaki en büyük sayı bulup bir bağlı listeye (**dlist**) ekleyecektir. Okuma işlemi dosya sonuna kadar yapıp tamamlandığında, sayılar listeden ters sırada ekrana bastırılacaktır.

**Not:** Tanımlanan işlemi C dilinde ve **libfdt** kütüphanelerini kullanarak yazınız, **libfdt** fonksiyonları kullanılmadan yapılan çözümlere puan verilmeyecektir.

## Soru 24

Aşağıda verilen kod parçalarından hangisi C için hatalıdır?

- A** ☐ `char *ch = "SisProg";  
printf("%c", ch);`
- B** ☐ `char ch[] = "SisProg";  
printf("%c", *ch);`
- C** ☐ `char *ch = "SisProg";  
printf("%c", ch[1]);`
- D** ☐ `char *ch = "SisProg";  
printf("%c", *++ch);`
- E** ☐ `char ch[7] = "SisProg";  
printf("%c", ch[5]);`

Seçimi Boş Bırakmak İstiyorum

**Soru 25**

Puan: 3,00

"**belge.pdf**" adlı bir dosyanın izinlerini; dosya sahibinin ve grubunun dosyayı düzenleyebilmesi ve sahip olmayan ve gruba ait olmayan kullanıcıların sadece görüntüleyebilir ancak değiştiremez olması gerekiyor. Aşağıdakilerden hangi Linux kabuk (shell) komutu bunu yapar?

**A** ☐ \$> chmod 641 belge.pdf

**B** ☐ \$> chmod 764 belge.pdf

**C** ☐ \$> chmod 664 belge.pdf

**D** ☐ \$> chmod 651 belge.pdf

**E** ☐ \$> chmod 554 belge.pdf

Seçimi Boş Bırakmak İstiyorum