

DENEY-1 (LINUX İŞLETİM SİSTEMİ)

ÖLÇÜSÜZ DENEY
≈ MİNE UĞAN ≈

1

Linux: Açık kaynak kodlu ve ücretsiz bir işletim sistemi cehridir.

Bu cehirdeki tamamlanan işletim sistemi (Linux Dağıtımı) ortaya çıkar.

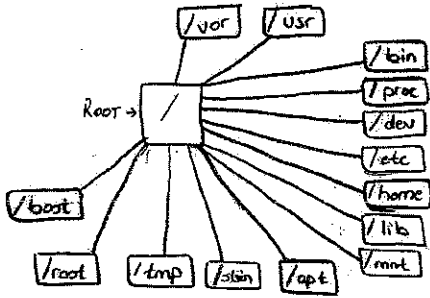
Linux Dağıtımları ⇒ Ubuntu, Pordus, Debian, SUSE, Red Hat gibi

Avantajları

- ⇒ IP adresi değiştirme, yönlendirme tablosu oluşturma, qd trafiki gibi işlemler için basit ve güçlü ortam sağlar.
- ⇒ Server / Client uygulamalarının oluşturmaları ve çalıştırması basittir.
- ⇒ Basit bir şekilde Router olabilir.
- ⇒ Ap sunucuları, qd cihazları ve gömülü sistemlerde yaygındır.
- ⇒ Kullanılan yazılımlar korunan değildir.

Linux Kullanıcı Hesapları ve Dosya Hiyerarşisi

- * Root ⇒ En yetkili kullanıcıdır. Super-user ile denmektedir.
- Root yetkisi gerektiren işlemlerde "sudo" komutu kullanılır.
- * Tüm klasörlerin altında kök dizini olan root klasörü bulunur.
- * Root dizini başlangıçta kök olarak "/" simgesiyle gösterilir.



* /root başlangıçta dizinlerde sonra bu klasörler gösterilir.

Dosya Komutları

- cat ⇒ Bir dosyayı okuyup gibi ekrana yazdırmak
- cp ⇒ dosyaların kopyalanması.
- rm ⇒ dosyaların silinmesi.
- mv ⇒ dosyaların taşınması.
- ls ⇒ dosyaların listelenmesi.
- mkdir ⇒ dosya oluşturmak

* pwd ⇒ Dosya sisteminde nerede olduğumuzu gösterir.

* whoami ⇒ Kim olduğumuzu sorar.

NOT: Sisteme aynı anda hangi kullanıcıların bağlanmış olduğunu; users, who, w komutları kullanılır.

* Kapatma komutları: halt - init0 - poweroff - shutdown } Bu komutlar için super-user olmak gerekir.

* Yeniden başlatma komutları: init6 - reboot

* Dosya Oluşturma ⇒ cat - touch - nano - vi → metin editörüdür.

* Bir dosya yada dizinin izinlerinin değiştirilmesi için chmod komutu kullanılır.

ÖRNEK: test.sh isimli script oluşturulalım;

```
$ chmod +x test.sh
```

```
$ ./test.sh read kpmutuylo klavyelerden giriş alınır;
```

```
#!/bin/sh
```

```
echo "What is your name?"
```

```
read PERSON
```

```
echo "Hello, $PERSON"
```

} çıktısı ⇒

```
$ ./test.sh
```

```
What is your name?
```

```
Mine U
```

```
Hello, Mine U
```

```
$
```

Not: Sayı 1'den 10'a kadar for döngüsü kullanarak ekrana yazma;

```
for i in {1..10}
```

```
do
```

```
echo $i
```

```
done
```

DENEY-2
(Ağ Güvenliği ve Etik Saldırı)

②

Penetrasyon Testi ⇒ Bilgisayar sistemin yada sistemin içeriden yada dışarıdan gelebilecek saldırılara karşı ne kadar güvende olduğunu konuyla ilgili test edilir.

⇒ Sistemin güvenli olmasını sağlamak amacıyla acıkarlarının önlenmesidir.

⇒ dos, application, internet, intranet, process gibi bölümler içerir.

DOS ⇒ Şirketin herhangi bir saldırıda hizmetlerinin çöküp çökmediği test edilir.

Application ⇒ Uygulama kumanda kullanan uygulamalar ve kodlarında acık taraması yapılır.
↳ İlgili Ağ durumunun testinde burada yapılır.

Intranet ⇒ Dış ağın penetrasyon testinde kullanılır.

↳ Dış dünyadan gelecek herhangi bir tehdide karşı durumu tespit edilir.

Avantajları

ve **process**

- * Saldırıya karşı daha dirençli bir alt yapı
- * Kullanıcı bazlı bilgi güvenliği sağlama
- * Sistem durdurulması ve kaynak doldurma engellenmesi
- * Veri olarak uyum sağlama
- * Kurum prestiji ve marka değerini koruma
- * BT kaynaklı risklerin azaltılması

8 terim ile anılır; Siyah - Beyaz - Gri Şapka diye;

Beyaz Şapka ⇒ İyi niyetli olan ve pentest yapan hackerlardır.

Siyah Şapka ⇒ Kötü niyetli olup crackers olarak adlandırılır.

(Kredi kartı bilgilerini çalan, firma sırlarını çalmak gibi)

Gri Şapka ⇒ Beyaz ve Siyah şapka ile bir kombinasyondur.

Herşeyi bilip yapabildikleri ve neyi okudukları için hackerler.

Penetrasyon Testinin Adımları

1. Bilgi Toplama ⇒ Sistem hakkında ön bilgi edinmek.
Internet, köpik bilgileri, DNS, IP, web sitesi yoluyla bilgi edinebiliriz.
2. Tarama ⇒ Sistemi banyo bilgi edinmek.
Sistemdeki bilgisayarlar ve network cihazlarının tespit edilmesi.
3. Erişim Sağlama ⇒ Sistemde bulunan acıklar kullanılarak sisteme izinsiz erişim sağlanması.
Yetki yükseltme, sistem durdurma.
4. Erişim Koruma ⇒ Elde edilen erişimin korunması.
5. İzleni Yok Etme ⇒ Hedef sistem üzerinde ilk 4 adımda yapılan işlemlerin bir kopyası izler temizlenir.

Ağ Güvenliği Test ve Denetim Araçları

- Ağ Dinleme Araçları
- Port Tarama Araçları
- Sifre kırma araçları
- Web Güvenliği Test araçları
- Genel Amaçlı Güvenlik Açığı Tarama Araçları

⊕ Ağ Dinleme araçlarının en sık kullanılan Wireshark'tir.
Wireshark ⇒ Açık kaynak kodlu bir uygulamadır.
Trafiki kaynak adres, hedef adres, kaynak port ve hedef port gibi kriterlere göre yakalar.
Ping ⇒ Özel bir Network potansiyelidir.

Nmap: Port taramacı programıdır.

③

Nmap Özellikleri

- * TCP ve UDP port taraması yapabilir.
- * İşletim Sistemi tespiti yapabilir.
- * Çalışan servisleri tespit eder.
- * Yazılımların sürümünü tespit eder.
- * Bir ağdaki canlı bilgisayarları tespit eder.

Nmap -p182.168.56.101 ⇒ Hedef mak. tüm portların taramasını sağlar.

WinPcap ⇒ Network kartlarına erişerek paket başında işlem yapmaya olanak sağlayan C dilinde yazılmış dll'dir. (dll → dinamik link library)

⇒ paket yakalama, ip izleme, protokol analizi, ip saldırı tespiti yapılır.

JnetPcap ⇒ Gerçek zamanlı yakalanan paketlerin criştilmesini sağlar.

⇒ Açık kaynak kodlu bir kütüphanedir.

TCP ve UDP'nin farkları

- TCP'de paketlerin karşı tarafa ulaşp ulaşmadığı kontrol edilir.
 - UDP'de bu kontrol sağlanmaz paketler direkt gönderilir.
 - TCP'de paketi gönderdikten sonra karşı taraftan cevap bekler.
 - UDP'de daha çok ses ve video paketi göndermede kullanılır. TCP'ye göre hızlıdır.
- fakat güvenli değildir.

Netsparker ⇒ Web sitesi uygulama seviyesindeki güvenlik açıklarını analiz eder.

⇒ Güvenlik Taramasıdır.

DENEY-3 (Bilgisayar Ağı Cihazlarının Yapılandırılması)

(4)

Router

- ilk açıldıklarında port işlemini yapar.
- CPU, hafıza, ana birim devreleri gibi sistem donanımlarını kontrol eder.
- Üzerlerinde, IOS (Internetwork Operating System) işletim sistemi çalışır.
- Bu sistemde iki farklı mod vardır.

⇓ ⇓
User exec Privileged exec

User exec modu; Router'a ilk login olduğumuzda bu moda olur.
Sadece bilgi görüntüler, konfigürasyon değişikliği yapılamaz.

Privileged exec moduna; konfigürasyon değişikliği yapılır. User modtan bu moda geçmek için "enable" komutu kullanılır.

Hypterminal

⇒ Ağı cihazlarını konfigüre etmek için kullanılan bir terminal emülasyon yazılımıdır.

⇒ Router üzerinde konfigürasyon değişikliği yapmak için;

RIP için ;

```
Router # config term
Router (config) # router rip          ↳ bizim fast
Router (config-router) # network 192.16.0.0
Router (config-router) # network 192.17.0.0    ↳ bizim serial
Router (config-router) # exit
Router (config) # exit
Router # ping 192.16.0.1
↳ karşı taraf fast
```

DTE

```
Router > enable
Router # configure terminal
Router (config) # interface fast 0/0
Router (config-if) # ip address 192.16.0.1 255.255.0.0
Router (config-if) # no shutdown
Router (config-if) # exit
Router (config) # interface serial 0/1
Router (config-if) # ip address 192.17.0.1 255.255.0.0
Router (config-if) # no shutdown
Router (config-if) # exit
```

DCE

```
Router > enable
Router # configure terminal
Router (config) # interface fast 0/0
Router (config-if) # ip address 192.18.0.1 255.255.0.0
Router (config-if) # no shutdown
Router (config-if) # exit
Router (config) # interface serial 0/0
Router (config-if) # ip address 192.17.0.2 255.255.0.0
Router (config-if) # clock rate 8000
Router (config-if) # no shutdown
Router (config-if) # exit
```

DENEY - 4 (MOBİL UYGULAMALAR)

5

Bir Android Projesinin Yapısı;

src ⇒ Kaynak dosyalar bulunur. Java dosyaları tutulur.

res ⇒ Kaynak kod dışında; resimler, videolar .. gibi görselleştirme herşey burada tutulur.

drawable ⇒ Resim dosyalarını içerir.

layout ⇒ Ekranın bir tasarım dosyası bulunur. onCreate metodunu çağırır.

menu ⇒ Cihazın menü tuşuna basıp menü çıkmasını istiyorsanız xml dosyasını burada tanımlarsınız.

value ⇒ Uygulamada kullanılan sabit değişkenlerdir. strings.xml dosyasındaki metinler saklanır.

AndroidManifest.xml ⇒ Uygulamanın kolbidir. Bütün ihtarlar burada yapılır. Uygulama içindeki ekranları burada kaydedilip tanımlanmış zorunludur.

⇒ <intent-filter> tagi ile activity sayfasının ilk çalıştırılacağı activity belirtilir.

⇒ ihtiyacı duyulan API kütüphaneleri belirtilmektedir.

String.xml ⇒ Farklı stil ve formatta global erişilebilen string değişkenlerinin tanımlandığı yerdir.

Activity Sınıfı ⇒ Android programlarının en temel bileşendir. Her bir ekran bir activity'dir.

⇒ Activityler kullanılabilmek için manifest dosyasında tanımlanmalıdır.

Activity Tanımlama Örneği;

```
<manifest ...>
```

```
<application ...>
```

```
<activity android:name=".ornekactivity" → java sınıfının adı
```

```
android:label=".ornekactivity" → activity'nin uygulamada görünen ismidir.
```

```
</application ...>
```

```
</manifest ...>
```

Intent Sınıfı ⇒ Bir activityden diğeriye geçmek için intent sınıfı kullanılır.

Bir activity'nin başka bir activity çağırması;

```
Intent intent = new Intent (this, SignInActivity.class);
```

```
startActivity (intent);
```

Intent-filter ekleme;

```
<activity android:name="ShareActivity">
```

```
<intent-filter>
```

```
<action android:name="android.intent.action.SEND"/>
```

```
<category android:name="android.intent.category.DEFAULT"/>
```

```
<data android:mimeType="text/plain"/>
```

```
<data android:mimeType="image/*"/>
```

```
</intent-filter>
```

```
</activity>
```

(NOT): Tanımlanan her <intent-filter>

inde <action>, <category>

<data> tanımlamalarını

birlikte fotoğraflar

• Layout → Her ekrana ait tasarımları barındırır;

Layout örneği;

(6)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill-parent"
    android:layout_height="fill-parent"
    android:orientation="vertical">
    <TextView android:id="@+id/text"
        android:layout_width="wrap-content"
        android:layout_height="wrap-content"
        android:text="Hello, I'm a TextView"/>
    <Button android:id="@+id/button"
        android:layout_width="wrap-content"
        android:layout_height="wrap-content"
        android:text="Hello, I'm a Button"/>
</LinearLayout>
```

* Bu layout res/layout/main-activity.xml kayıtlıktan sonra Activity'de onCreate() metodunda kodun yazılması gerekir;

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main-activity);
}
```

DENEY - 5 (GEMÜLÜ SİSTEM PROGRAMLAMA)

7

Gemülü Sistem ⇒ Daha büyük bir sistem içinde belirli kontrol işlevleri için tasarlanmış bilgisayar sistemidir.

⇒ Gemülü cihazın kontrolü, izlenmesi veya çalışmasına yardımcı olur.

⇒ Kendisi için önceden özel olarak tanımlanmış görevleri yerine getirir.

⇒ Gemülü sistemin çerçevesini; mikroisimleri ya da mikrodeneğiçiciler oluşturur.

Örnek olarak ⇒ Banka ATM'leri

- Cep Telefonları
- Router (Yönlendirici), timeserver ve güvenlik duvarı (firewall)
- Bilgisayar yatakları
- Hesap mak.
- Mikrodüğü Fırın, Gamaşır mak, TV setleri...

Özellikleri

- * Genellikle ya hiçi ya da kısıtlı kullanıcı ona göre sahiptir.
- * Verimlilik önemlidir. Maliyet optimize edilir.
- * Sürekli olarak yürütülür.
- * Gerçek zamanlı kısıtlamaları koruyarak işare tasarlanmıştır.
- * Yazılım ve donanım olarak 2 kısımdan oluşur.

Gemülü Sistem Programlama ve Havuüsti Uygulamaların Faktörleri

- ⇒ Gemülü cihazlar sınırlı kaynağa sahiptir.
- ⇒ Gemülü sistem ile bilgisayarcı kullanan bileşenler farklıdır.
- ⇒ Gemülü sistem tipik olarak daha küçük ve daha az güç tüketen bileşen kullanır.
- ⇒ Gemülü sistem, daha fazla donanıma bağlıdır.

Gemülü Sistemde hedef : Minimum Abn ve minimum zamanda max. özellikli program elde etmektir.

Gemülü Sistemde Kullanan Diller : Machine kodu

Düşük sev. Dil : Assembly

Yüksek sev. Dil : C, C++, Java, Ada

Uygulama Düzeyi : Visual Basic, Access

} Temel Kullanan Dil = C'dir.

ARDUINO : Gemülü Sistem Prog. için kullanan araçların birisidir.

Elektronik prototip geliştirme araçlarıdır.

Arduino Uno, mega, mega ADK, LilyPad çeşitleri bulunur.

Arduino

- * Kolay şekilde çevresine diğer açık kaynaklı geliştirme platformudur.
- * Üzerinde Atmega firmasının 8 ve 32 bit mikrodeneğiçicileri bulunur. (Atmega328)
- * Arduino kütüphaneleri ile mikrodeneğiçicilerle programlanır.
- * Analog veriyi, Dijital veriyi kolaylıkla işler.
- * Sensörlerden gelen veriyi kullanabiliriz.
- * Dış dünyaya çıktıları üretebiliriz.
- # Arduino yazılım programları Sketch denmektedir.

* Arduino programları temel 2 kısımdan oluşur. Bunlar setup() ve loop()'dur.

Setup() ⇒ Board ile ilgili ayarlamalar yapılır.

Program ilk yükledikten sonra icra edilir.

Loop() ⇒ programda sürekli icra edilecek bölümler yer alır.

* 13 numaralı bir led 1 sn aralıklarla yakıp söndüren program;

```
void setup() {  
    pinMode(13, OUTPUT);    → pin 13 çıkış yapıldı.  
}  
void loop() {  
    digitalWrite(13, HIGH);    → High konumunda 13 numaralı led yanar.  
    delay(1000);                → 1 sn bekle.  
    digitalWrite(13, LOW);     → Low konumunda 13 led söner.  
    delay(1000);                → 1 sn bekle.  
}
```

* LCD kayan isim;

```
#include <LiquidCrystal.h>  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  
void setup() {  
    lcd.begin(16, 2);  
}  
void loop() {  
    for (int i=0; i<17; i++) {  
        lcd.setCursor(i, 0);  
        lcd.print("mine");  
        lcd.setCursor(0, 1);  
        lcd.print("ucun");  
        delay(500);  
        lcd.clear();  
    }  
}
```

* 8 bit sayıcı;

```
int butondeperbasla = 0;  
int butondeperbitir = 0;  
int count = 0;  
int state = LOW;  
void setup() {  
    pinMode(0, OUTPUT);  
    pinMode(1, OUTPUT);  
    pinMode(2, OUTPUT);  
    pinMode(3, " ");  
    pinMode(8, INPUT);  
    pinMode(9, INPUT);  
}  
void loop() {  
    butondeperbasla = digitalRead(8);  
    butondeperbitir = digitalRead(9);  
    if (butondeperbasla == HIGH) {  
        count++;  
        sayici();  
    }  
    if (butondeperbitir == HIGH) {  
        ...  
    }  
}
```

* LCD'de yazan sayıların EBOB'unu bulan program;

```
#include <LiquidCrystal.h>  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  
int sayi1 = 10;  
int sayi2 = 5;  
void setup() {  
    int sayi1 = 10;  
    int sayi2 = 5;  
    lcd.begin(16, 2);  
    lcd.setCursor(0, 0);  
    lcd.print(sayi1);  
    lcd.print(" ");  
    lcd.print(sayi2);  
    int sayac;  
    int sonuc;  
    if (sayi1 > sayi2) {  
        sayac = sayi1;  
    } else {  
        sayac = sayi2;  
    }  
    while (true) {  
        if ((sayi1 % sayac == 0) && (sayi2 % sayac == 0)) {  
            sonuc = sayac;  
            break;  
        }  
        sayac--;  
    }  
    lcd.setCursor(0, 1);  
    lcd.print(sonuc);  
}  
void loop() {  
}
```


Dağıtık Sistem ⇒ Katmanların birbirinden uzak yerlerde bulundukları sistemlere denir.

⇒ Farklı bilgisayarlara ait donanım ve yazılım bileşenleri arasında yalnızca mesajlaşma yöntemleri kullanılarak haberleşme ve koordinasyonun sağlandığı network sistemleridir.

⇒ Bağımsız bilgisayarların kullanıcılara tek bir bilgisayar gibi görünerek iş yükünün paylaşıldığı sistemlerdir.

⇒ TCP ve UDP en çok kullanılan mesaj transfer protokolleridir.

Dağıtık Programlama da;

Kaynak Paylaşımı: Bir sitedeki kullanıcı, diğer sitede mevcut kaynakları kullanması mümkündür.

Hesaplama, Hızlandırma: Dağıtık sistem farklı siteler arası ott hesaplamaları yapar. Eş zamanlı çalıştırılabilir ve hesaplamaları hızlanır.

Güvenlik: Dağıtık sistemde bir site başarısız olursa, diğer sistemler etkilenmez. Güvenilir sistem veriyi devam ettirebilirler.

İstemci ⇒ Hizmetler için istekte bulunmaktadır.

Sunucu ⇒ İsteğe bulunan hizmetleri isteret yanıt verir.

MD5 Algoritması ⇒ Tek yönlü bir işleme algoritmasıdır.

Giriş karakterlere göre (terabyte 40 db düşüp olso) 128 bit bir çıktı üretir.

Java'da Socket İşlemleri

* Ap üzerinde birbirine bağlı bilgisayarlar arasında haberleşme portlar üzerinden yapılmaktadır.

* Bilgisayarlarda 16 bitle ifade edilebilecek sayıda ($2^{16} = 65.536$) port bulunmaktadır.

* İlk 1024 3 özel amaçla ayrılmıştır.

* İlk kullanılmayıp port → 1024, son portumuz ise = 65.536. porttur.

⊗ `ServerSocket socket = new ServerSocket (PORT-NUMARASI);` ⇒ SOCKET nesnesi oluşturma.

⊗ `Socket bağlantı = socket.accept();` ⇒ Belirtilen port numarasını dinleyerek yeni bağlantı bekler.

⊗ `Socket bağlantı = new Socket (IP-ADRESİ, PORT-NUMARASI);` ⇒ Diğer tarafta server tarafından açılan porta bağlanmak için.

NOT: Kendi bilgisayarında açılan porta erişmek için

IP-ADRESİ yerine "localhost" yazılır.

⇒ Server ve Client arasındaki bağlantı kurulduktan sonra her iki taraf da socket nesneleri (bağlantı) yardımıyla veri alışverişi yapılabilir.

Veri Gönderimi

```
PrintWriter output = new PrintWriter(bağlantı.getOutputStream(), true);  
output.println ("Gönderilecek Mesaj");
```

Veri Alımı

```
BufferedReader in = new BufferedReader (new InputStreamReader (bağlantı.getInputStream()));  
in.readLine();
```

NOT: `in.readLine` satır gelmesi için bekler, bir sonraki adıma geçmez.

Bekleme istemiyorsak bunun önce `in.ready()` fonk. edilebilir.

Java'da Thread İşlemleri;

* In.ReadLine komutu bir sonucu okuma işlemi yapmayı beklediğinden birden fazla bilgisayara aynı anda bağlanmak için Thread kullanılır.

* Thread Örneği;

```
public class örnek extends Thread {  
    public void run () {  
        int i=0;  
        while (true) {  
            System.out.println (i++ + " ");  
        }  
    }  
    public static void main (String args[]) {  
        (new örnek()).start();  
        (new örnek()).start();  
    }  
}
```

DENEY - 7 (GÖRÜNTÜ İŞLEME UYGULAMALARI)

(11)

Sayısal Görüntü İşleme ⇒ Elde edilen görüntünün bilgisayarla okutulup bir işlem yapılması ve ardından görüntüyü çıkışa iletilmesidir.

⇒ Teknikleri ile ; Sayısal görüntüler iyileştirilerek nesne tanıma, hedef tanıma işlemleri gerçekleştirilir.

⇒ Tasarım, imalat, güvenlik, tıp, elektronik, makine ve jeodezi ... alanları.

Amaçları ⇒ Görüntü iyileştirme

⇒ Görüntü sıkıştırma

⇒ Biyometrik Tanıma

⇒ Otomatik yüz aracı obje tanıma ve takip etme.

İkili Görüntü (Binary Image) :

* Siyah - Beyaz resimdir.

* Siyah - Beyaz resim sadece iki gri değerden oluşur. (Her pixel ya siyah ya beyaz'dan oluşur.)

* Beyaz pixeller $\rightarrow 1$, Siyah pixeller $\rightarrow 0$ ile gösterilir.

* Renkli görüntüler bilgisayarda 24 bitlik veri olarak görüntülerir.

* Görüntüleme R - G - B 3 adet gri düzeyli görüntünün üst üste gelmesiyle oluşur.

Niteliklendirme

* Görüntünün pixel değerinin belli aralıklarda olması, görüntünün niteliğini belirtir.

* $b=8 = 2^8 = 256$ adet gri tonu bulunmaktadır.

Matlab Görüntü İşleme

imread ⇒ Bu komut ile fotoğraf matlab workspace için tanımlanır.
Ve ardından imshow ile pencerede gösterilebilir.

imshow ⇒ İşlemler için pencere açılır.

`I = imread('ornek.jpg');`

`imshow(I);`

`imshow(I);`

(NOT) : Verileri temizlemek için ; clear, all ve clc komutları en başa yazılmalıdır.

warped ⇒ Görüntüyü belli yüzeylerde gösterir.

`resim = imread('rice.png');`

`[x1,y1,z1] = cylinder;`

`[x2,y2,z2] = sphere;`

`imshow(resim); title('Original resim');`

`figure;`

`subplot(1,2,1); warped(x1,y1,z1,resim); title('cylinder');`

`subplot(1,2,2); warped(x2,y2,z2,resim); title('sphere');`

TÜRE DÖNÜŞÜM KOMUTLARI

(12)

rgb2gray ⇒ renkli görüntüyü gri seviyeye çevirir.

graythresh ⇒ Görüntüdeki portlaklık eşikini otomatik olarak belirler.

* Görüntü üzerinde belirli işlem yapabilmemiz için gri seviyede çalışmamız gerekir;

I = rgb2gray(I) komutu kullanılır.

```
level = graythresh(I)
```

```
bw = im2bw(I, level)
```

```
bw = bwareaopen(bw, 50)
```

```
figure; imshow(bw);
```

im2uint8 ⇒ uint8 renk hassasiyetine dönüştürür.

demosaic ⇒ Hermetik görüntüyü renkli görüntüye çevirir.

gray2ind ⇒ Yönlü görüntüyü endeksli görüntüye çevirir.

gray2ind ⇒ Eşik departerle yönlü görüntüsünde eşikli görüntü oluşturma.

im2int16 ⇒ Görüntüyü 16 bitlik tam sayılara dönüştürme.

label2rgb ⇒ Etiket matrisini RGB'ye dönüştür. (renkli)

im2double ⇒ Resmi çift hassasiyete dönüştür.

im2uint16 ⇒ Görüntüyü 16 bite dönüştür.

mat2gray ⇒ Resmi yönlü görüntüsüne dönüştür.

GÖRÜNTÜ ANALİZİ KOMUTLARI

bwboundaries ⇒ Binary maskin bölgelerinin sınırlarını belirler.

* Basit Mesajı sayma uygulaması;

```
I = imread('rice.png');
```

```
BW = im2bw(I, graythresh(I));
```

```
B = bwboundaries(BW);
```

```
figure; imshow(BW);
```

```
text(10,10, strcat('\color{red} bulunan sınırlar : ', num2str(length(B))))
```

```
hold on;
```

```
for k=1:length(B)
```

```
    boundary = B{k}
```

```
    plot(boundary(:,2), boundary(:,1), 'g', 'linewidth', 4)
```

```
end
```

Edge ⇒ Özel filtreler yardımıyla gri seviyedeki görüntülerin sınırları belirlenir.

```
I = imread('rice.png');
```

```
imshow(I);
```

```
BW1 = edge(I, 'prewitt');
```

```
BW2 = edge(I, 'canny');
```

```
figure; imshow(BW1);
```

```
figure; imshow(BW2);
```

Diğer Analiz Komutları ⇒ Hough, houghlines, corner...

Görüntü İyileştirme Komutları

(13)

imadjust ⇒ Görüntü yopunluluğu değerini ve renk haritasını ayarlar.

histeq ⇒ Histogramı eşitleyerek kontrastı artırır.

adapthisteq ⇒ CLAHE algoritmasını kullanarak kontrastı artırır.

↳ CLAHE ⇒ Kontrast sınırlı adaptif histogram eşitleme algoritmasıdır.

```
I = imread('tire.tif');
```

```
A = adapthisteq(I, 'clip Limit', 0.02, 'Distribution', 'gaussian');
```

```
B = histeq(I);
```

```
subplot(231), imshow(I);
```

```
subplot(232), imshow(A); title('adapthisteq uygulanmış');
```

```
subplot(233), imshow(B); title('histeq uygulanmış');
```

Görüntü iyileştirme yapılmıştır.

medfilt2 ⇒ 2 boyutlu medyan filtreleme yapar.

wiener2 ⇒ 2 boyutlu adaptif görüntü temizleme filtresidir.

Bölgelerin Belirlenmesi

bwboundaries ⇒ Sınırları ile çizilir.

plot ⇒ Çizimin kalınlığı, rengi istenilen şekilde belirlenir.

Alanların Hesaplanması ⇒ **bwlabel**: Bağlı bileşenlerin bilgileri bir depiştirece atar.

regionprops ⇒ Bölgelerin özellikleri çıkarılır.

Fork Hesaplama

```
im1 = rgb2gray(imread('resim1.png'));
```

```
im2 = rgb2gray(imread('resim2.png'));
```

```
fork = imabsdiff(im1, im2);
```

```
bw = bwisopen(fork, 55);
```

```
bw = imfill(bw, 'holes');
```

```
SE = strel('square', 1);
```

```
bw2 = imerode(bw, SE);
```

```
fork = regionprops(bw2, 'all');
```

```
C = [fork, Centroid];
```

```
imshow('resim2.png');
```

```
title(['Toplam fork: ', num2str(length(fork))]);
```

```
hold on;
```

```
x = C(1:2:end);
```

```
y = C(2:2:end);
```

```
plot(x, y, 'yo', 'MarkerSize', 20, 'linewidth', 4);
```

Kinect Teknolojisi

* Sensör, son derece algılama sensör donanımları içermektedir.

→ Derinlik kamera sensörü

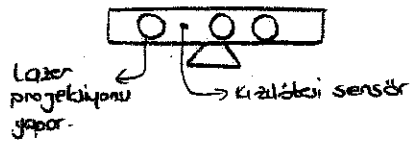
→ RGB kamera sensörü

→ Dörtü mikrofon grubu

→ 3 boyutlu hareketi yakalama, yüz tanıma, ses algılama

* Kinect 40-tıllım Geliştirme kiti bir iskelet, insan vücuduna göre numaralandırılmış 20 adet eklemden oluşur.

* Bu eklemler ve pozisyonları görüntüye iletir.



3 adet göz, 5 no mikrofonlar ve hareket sepiyacı motor mekanizması bulunur.

* `joint kafa = skeleton.joints[jointsType.Head];` → Kafa Tanımlanması

`joint sol el = skeleton.joints[jointsType.HandLeft];` → Sol el Tanımlanması

x ve y koordinatına göre kontrol yapılır;

* `if (-30 > handLx && handLx > 70 && -13 < handLy && handLy > 27)`

{
 gibi if şartları içinde tanımlama yapılır.

{
 `hareket-content = "hareket tamam";`
}

else {
 `hareket-content = "hareket algılanmadı";`
}

① HTTrack, Harvester, Medusa, nmap-p-192.168.56.101, fping-a-g-192.168.56.101 192.168.56.254

Komut ve araçların ne işe yaradığını araştırınız.

Cevap: HTTrack ⇒ Sitelerin kopyalarının indirilmesini sağlar. Bu sayede offline araştırma yapılabilir.

Harvester ⇒ Bir domaine ait alt domain ve e-mail hesaplarını listeler.

Medusa ⇒ Belli kullancı adı üzerine şifre kombinasyonları dener.

Nmap-p ⇒ Hedef ip açık port taraması yapar.

Fping-a-g ⇒ ip aralığındaki pc'lere ping atarak açık olanları listeler.

② İnternete bağlı iki bilgisayardan biri client diğeri de server olarak kullanılmaktadır. Client'dan servere mesaj gönderilebilmesi için client tarafından yazılması gereken kod parçasını yazınız. (ip ve port no rastgele)

Cevap:

```
public class TCPCClientLocal {
    public static void main (String[] args) {
        Socket link = null;
        try {
            link = new Socket ("192.168.56.101", 9000);
            PrintWriter output = new PrintWriter (link.getOutputStream(), true);
            output.println (message);
        } catch (IOException e) {}
    }
}
```

③ Ağ yapısını kullanarak Router A ve Router B konfigürasyonları ile RIP yönlendirme algoritmasını kullanarak yapılandırma komutları yazınız.

Router A ⇒ Serial 0 Adres , Serial 1 Adres , Subnet mask
201.100.11.1 ————— 255.255.255.0

Cevap: Router B ⇒ ————— 201.100.11.2 , 255.255.255.0

Router A

```
# config terminal
# interface fast 0/0 (so)
# ip address 201.100.11.1 255.255.255.0
# no shutdown
# interface serial 0/1
# ip address 192.7.7.1 255.255.255.0
# no shutdown
# exit
# config terminal
# router rip
# network 192.7.7.0 → ethernet adresi
# network 192.7.7.0 Sonu sıfır olamaz.
# network 201.100.11.0
# exit
# exit
# exit
```

Router B

```
# config terminal
# interface serial 0/0
# ip address 201.100.11.2 255.255.255.0
# no shutdown
# exit
# config terminal
# router rip
# network 201.100.11.0 → ethernet sonu 0 olamaz
# network 201.100.11.0
# exit
# exit
```

4) Aşağıdaki Soruları Cevaplayınız.

2

a) Gerçek (absolute) / Göreceli (relative) dizin nedir?

Cevap: Gerçek dizin: Kalk dizinden itibaren dosyanın adresini gösterir. Her yerde çalıştırılabilir. (/var/html/index.html)

Göreceli dizin: içinde bulunan dizine göre hesaplanan kesin olmayan dizindir. (/html/index.html)

b) \$ls -l

drwxrwxr-x 1 admin haroid-finch 5344 Dec 29.08.38

Sohin-Reese.jpg

yukarıdaki \$ls -l komutundan sonra gösterilen d, r, w, x harflerinin anlamlarını yazınız.

Cevap: d == dizin

ilk drwx = kullanici dizini

r = okunabilir

ortu rwx = kullaniciin dahil okuyup grup dizini

w = yazılabilir

son rwx = ortak izimler herkes.

x = çalıştırılabilir.

c-) Bir dizinde bulunan sonu ".sh" ile biten dosyaların isimlerini "1.txt, 2.txt ... n.txt" şeklinde depistiren shell programı yazınız.

Cevap #! bash/sh

echo "Dizin gir"

read yol

for i in \$(ls \$yol); do

mv -- "\$i" "\${i%/*}/basename \$i.sh '/' print ((\${1var++}))".txt }

done

d-) Kullanıcıdan şifre girişini yapmasını isteyip, "unixstudent" ise ekrana "şifre doğru" değilse "şifre yanlış" yazan shell programı yazınız.

Cevap: #! bash/sh

echo "şifre giriniz."

read SIFRE

if ["\$SIFRE" = "unixstudent"]; then

echo "şifre doğru"

else

echo "şifre yanlış."

fi

e-) Amazon EC2 üzerinde ücretsiz bir sunul makina çalıştırıldığında public ve private ip adresinden hesaplarının verilmesi gerekir? neden?

Cevap: Amazon EC2 sunul bilgisayarların uzaktan güvenli bağlantıyla (SSH) kontrol edildikleri için internete çıkabilen bir public IP'ye sahip olması gerekir.

deper=0

for i in \$(ls *.txt); do

deper=\$((deper+1))

mv \$i \$deper.txt

done

⑤ * KINECT ÖNEMLİ TANIMLAMA * *

③

```

joint solEl = skeleton.joints [ jointType. HandLeft ];
joint soldirsek = skeleton.joints [ jointType. ElbowLeft ];
joint solOmuz = skeleton.joints [ jointType. ShoulderLeft ];
joint omurga = skeleton.joints [ jointType. Spine ];
joint kafa = skeleton.joints [ jointType. Head ];
joint sapAyakBilet = skeleton.joints [ jointType. AnkleRight ];
joint solAyakBilet = skeleton.joints [ jointType. AnkleLeft ];
joint sapOmuz = skeleton.joints [ jointType. ShoulderRight ];
joint sapEl = skeleton.joints [ jointType. HandRight ];
joint sapDirsek = skeleton.joints [ jointType. ElbowRight ];
joint boyun = skeleton.joints [ jointType. ShoulderCenter ];
joint sapayak = skeleton.joints [ jointType. FootRight ];
joint solayak = skeleton.joints [ jointType. FootLeft ];

double solElx = solEl.position.x * 100;

```

⊗ sap-sol el, sap-sol dirsek, kafa, sap-sol omuz 'ların aynı hızda olduğunu kontrol eden 8 koşullu if yapıları.

```

if ( -30 > handLx && handLx > -70 && -13 < handLy && handLy < 27 )
{
    if ( handRx < 31 && handRx > -9 && handRy < 14 && handRy > -26 )
    {
        if ( ElbowRx < 20 && ElbowRx > -12 && ElbowRy < 20 && ElbowRy > -2 )
        {
            if ( ElbowLx < 13 && ElbowLx > -21 && ElbowLy < 19 && ElbowLy > -3 )
            {
                if ( ShoulderLx < 20 && ShoulderLx > -4 && ShoulderLy < 25 && ShoulderLy > -5 )
                {
                    if ( ShoulderRx < 19 && ShoulderRx > -10 && ShoulderRy < 35 && ShoulderRy > -14 )
                    {
                        hareket.content = "hareket tamam";
                    }
                }
            }
        }
    }
}
else
{
    hareket.content = "hareket iptal edildi.";
}
}

```

- 6) 0-7 arasında onluk tabanla verilen bir sayıyı ikilik tabana çeviren ve çevirmiş olduğu ikilik sayıya NOT (Değil) işlemini uygulayarak seven segment displaye gönderen C prog. veriniz.

(4)

Örneğin sayı=6 → 110 → Değili = 001 olarak.

```
Cevap: int sayi=9;
int dizi[5];
int j,k,q,t;
void setup() {
  int i; j=0, k=0, q=0, t=0;
  for (i=0; i<12; i++) {
    pinMode(i, OUTPUT); }
  //ikilik Taban
  do {
    dizi[j] = sayi % 2;
    sayi = sayi / 2;
    j++; }
  while (sayi >= 2);
  dizi[j] = sayi;
  for (k=0; k<sizeof(dizi); k++) {
    if (dizi[k] == 0) {
      dizi[k] = 1; }
    else {
      dizi[k] = 0; } } }
```

```
void loop() {
  //led display yaz.
  q=0; t=0;
  for (t=0; t<sizeof(dizi); t++) {
    if (dizi[t] == 0) {
      digitalWrite(g, LOW);
      digitalWrite(++g, LOW);
      digitalWrite(++g, LOW);
      digitalWrite(++g, LOW); }
    else {
      digitalWrite(g, LOW);
      digitalWrite(++g, LOW);
      digitalWrite(++g, LOW);
      digitalWrite(++g, HIGH); }
    q++; } }
```

- 7) a) Ekranla "Merhaba Dünya" yazan shell programı;

```
#!/bin/bash
echo "Merhaba Dünya"
```

- b) Bir kullanıcının tüm dosyalarını yedekleyen komut yazınız.

```
#!/bin/bash
echo -cuf mine-yedek.tar /home/mineuser*
```

- c) Disandan bir dizin parametre olup bu dizinin var olup olmadığını ekrana yazdırma.

```
#!/bin/bash
echo "Ardığınız dizini girin"
read dizin
if [-d $dizin];
then
  echo "dizin mevcut"
else
  echo "dizin mevcut değil"
fi
```

- d) Ekrana tarih -soat, kullanıcı adı, içinde bulunan dizini yazan shell prog.

```
→ date (tarih-soat)
→ whoami (kullanıcı kim)
→ pwd (nerdesin)
```

- e) Kullanıcıdan şifre isteyerek şifre "unixstudent" ise doğru yazsın, değilse yanlış yazsın

```
#!/bin/bash
echo "şifre giriniz."
read parola
if [ $parola = "unixstudent" ];
then;
  echo "şifre doğru"
else
  echo "şifre yanlış"
fi
```

⑧ @Amazon cloudbe oluşturdığımız soru mak java ve python dili ile merhaba Dünya yazdırma;

⑤

java {
→ sudo touch merhabo.java → java klasiği olusturuldu.
→ sudo nano merhabo.java → klasiğin icine girip merhabo dunya yazıldı.
→ javac merhabo.java → klasiği derledi.
→ java merhabo → Derleneni calistirdi.
Hello, world
python {
→ sudo touch mine.py → python dosya olusturuldu.
→ nano mine.py → python dosyanin icieripi ocildi. Merhabo Dunya yotdik
Hello world

⑥ Sistemdeki kullanicilarin adini yotip, merhabo mine, ozon, Ali, Veli yotzen shell programi;

Cevap: cd odev
/odev : ./welcome.sh --help
kullanim : welcome.sh kullanicilar
/odev : ./welcome.sh mine, ozon Ali Veli
Merhabo mine
Merhabo ozon
Merhabo Ali
Merhabo Veli

⑨ Bir socket nesnesi olusturunuz. port numarasini dinleyerek yeni baglanti bekleyerek ve clipe
tarafita cliet tarafinda acilan porta baglanmak icin ne yapariz ?

Cevap: Serversocket socket = new Serversocket (PORT-NUMARASI); → socket olusturma.
Socket baglanti = socket.accept(); → port numarasini dinleyerek
yeni baglanti bekleme.
Socket baglanti = new Socket (IP-ADRESI, PORT-NUMARASI); → porta baglanma.

* Veri Gonderimi ve Veri Alimi yaptigimiz kod parcalarini yazalim;

Veri Gonderimi;

```
PrintWriter output = new PrintWriter (  
    baglanti.getOutputStream(), true);  
output.println ("Gonderilecek Mesaj");
```

Veri Alimi;

```
BufferedReader in = new BufferedReader (  
    new InputStreamReader (baglanti.getInputStream()));  
in.readLine();
```

- 10) ivme sensöründen y eksenini için gelen değerin 1 m/s^2 den büyük olup olmadığını 400 ms'de bir kontrol eden büyüklük "sayac" değişkenini bir artırarak ve "sayac" değerini "txtSayac" id'sine sahip bir TextView'de sürekli gösteren bir Android Uygulama.

(6)

Çözüm: public class MainActivity extends ActionBarActivity implements SensorEventListener {

Sensor accelerometer;

SensorManager sm;

TextView acceleration;

long bas;

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentLayout(R.layout.activity_main);

sm = (SensorManager) getSystemService(Context.SENSOR_SERVICE);

accelerometer = sm.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);

sm.registerListener(this, findViewById(R.id.acceleration);

bas = System.currentTimeMillis(); }

float yilk;

Static int sayac=0;

public void onSensorChanged(SensorEvent arg()) {

long son = System.currentTimeMillis();

if (son-bas >= 400) {

if (arg().sensor.getType() == Sensor.TYPE_ACCELEROMETER) {

float[] values = arg().values;

float x = values[0];

float y = values[1];

float z = values[2];

if (y-yilk >= 1) {

acceleration.setText(sayac++); }

yilk = y;

} } }

21) Sobel Görüntü Algoritması;

Çözüm: → Verilen herhangi bir resimdeki kenarları elde etmeye yarar.

→ Böylelikle resimler içindeki nesneler algılanıp gerekli işlemler yapılabilir.

→ İki adet konvolüsyon yöntemi vardır. Yatay kenar ve dikey kenar.

→ Bu yöntemler görüntü içerisindeki ışık yoğunluk değişiminin ani olduğu yerleri belirlememize yarar. (Bir nevi tırtıl yaktosisidir.)

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \Rightarrow \text{Yatay Sobel Kernel}$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \Rightarrow \text{Dikey Sobel Kernel}$$