

SELECT()

Select(), çeşitli dosya tanımlayıcıları üzerinde yapılacak I / O' nun var olup olmadığını ayırt etmek için işletim sistemini incelemenize izin veren bir sistem çağrısıdır.

Örneğin, **jtlnet** yazmak istediğinizi varsayalım. Bu “**telnet**” gibi çalışan bir programdır(çağı parçalar için). Kullanıcı bir sunucu ve port belirtir ve **jtlnet** sunucu ve port için bir **request_connection ()** gerçekleştirir.

Bağlantı yapıldığı zaman, **jtlnet** soket bağlantısından gelen herhangi bir baytı alacak ve standart çıktı üzerine bunları yazacak ve o standart inputtan herhangi bir giriş alacak ve onu soket bağlantısının üzerine yazacak.

Jtlnet soket bağlantısından ve **stdin** den okunan bayt'ların var olup olmadığını ayırt etmek için işletim sisteminin gerçekleştirilmesine ihtiyaç duyduğu için **select()**' e ihtiyaç duyar.

select()'in kullanışı kitabını okuyun. Bu çok güzel bir tanımlayıcıdır.

Select() için argümanlar aşağıdadır.

```
int select (int width, fd_set *readfds,  
            fd_set *writefds,  
            fd_set *exceptfds, struct timeval *timeout)
```

veri türleri **fd_set** ve **struct timeval**, **sys/types.h** ve **sys/time.h** 'ta tanımlanmaktadır. Bu sınıfta, biz istisnai durumların ve fd'nin yazılabilmesi için seçimleri dikkatli incelemeyeceğiz. Bu yüzden biz **select()**'i aşağıda olduğu gibi kullanacağız.

```
select(width, readfds, NULL, NULL, NULL);
```

Readfds dosya tanımlayıcısının bir bitmap'i dir. Bitmap'teki dosya tanımlayıcısının herhangi biri okumak için bir şeye sahip olduğu zaman **Select()** geri döndürecek.O geri döndürdüğü zaman, **readfds** bir şeyi okumayı sağlayan

tüm dosya tanımlayıcılarını içeren bir yeni bitmap olacak. Diğer bir deyişle, bu dosya tanımlayıcısı üzerindeki bir **read()** çağrısı kilitlemek için garantili değildir. O bir sıfır değeri ile geri dönebilir(yani dosyanın sonu).fakat o geri döndürmek için garantilidir.

Biz makrolarla bitmap'leri işledik.

FD_SET (fd, &fdset)

FD_CLR (fd, &fdset)

FD_ISSET (fd, &fdset)

FD_ZERO (&fdset)

Fd ,int tipinde ve fdset bir **fd_set**'tir. Bu yüzden dosya tanımlayıcısına sahip olan bir Bitmap yapmak için 0 ve 3 kurulmalıdır. Biz şimdi gerçekleştireceğiz:

```
fd_set fds;
```

```
FD_ZERO(&fds);
```

```
FD_SET(0, &fds);
```

```
FD_SET(3, &fds);
```

Biz **FD_CLR()** komutunu bit haritasında ki bitleri temizlemek için kullanırız ve **FD_ISSET()** komutunu belirli bitlerin set olup olmadığını test etmek için kullanırız.

Şimdi jtelnet(in **jtelnet.c**):

```
/* jtelnet.c
```

```
Jim Plank
```

```
CS360
```

```
Select Lecture
```

```
*/
```

```
#include "socketfun.h"
```

```
#include <stdio.h>
```

```
#include <sys/types.h>
```

```
#include <sys/time.h>
```

```
main(int argc, char **argv)
```

```
{
```

```
    int port;
```

```
    int fd;
```

```
    char *hostname, buf[100];
```

```
    fd_set readfds, masterfds;
```

```
    int nopen, n;
```

```
    if (argc != 3) {
```

```
        fprintf(stderr, "usage: jtelnet host port\n");
```

```
        exit(1);
```

```
    }
```

```
    hostname = argv[1];
```

```
    port = atoi(argv[2]);
```

```
    fd = request_connection(hostname, port);
```

```
FD_ZERO(&masterfds);
```

```
FD_SET(0, &masterfds);
```

```
FD_SET(fd, &masterfds);
```

```
while (1) {
```

```
    memcpy(&readfds, &masterfds, sizeof(fd_set));
```

```
    if (select(fd+1, &readfds, NULL, NULL, NULL) < 0) {
```

```
        perror("on select");
```

```
        exit(1);
```

```
    }
```

```
    if (FD_ISSET(0, &readfds)) {
```

```
        n = read(0, buf, 100);
```

```
        if (n == 0) {
```

```
            close(0);
```

```
            close(fd);
```

```
            exit(1);
```

```
        } else {
```

```
            write(fd, buf, n);
```

```
        }
```

```
    } else if (FD_ISSET(fd, &readfds)) {
```

```
        n = read(fd, buf, 100);
```

```
        if (n == 0) {
```

```
    close(0);

    close(fd);

    exit(1);

} else {

    write(1, buf, n);

}

}

}

}
```

İlk olarak **argv** ve **argc** beraber kullanılarak ve gerekli **request_connection** yapılır:

```
#include "socketfun.h"
```

```
#include
```

```
#include
```

```
#include
```

```
main(int argc, char **argv)
```

```
{
```

```
    int port;
```

```
    int fd;
```

```
    char *hostname, buf[100];
```

```
fd_set readfds, masterfds;

int nopen, n;

if (argc != 3) {
    fprintf(stderr, "usage: jtelnet host port\n");
    exit(1);
}

hostname = argv[1];
port = atoi(argv[2]);

fd = request_connection(hostname, port);
```

Şimdi biz iki **fd_sets** komutunu kullanacağız – **masterfds** olarak anılan bir sabit vardır,ve **readfds** olarak anılan bir geçici vardır.ilk olarak **masterfds** kullanıma sunulur bundan dolayı fd sıfır ve soket bağlantısı setlenir.

```
FD_ZERO(&masterfds);

FD_SET(0, &masterfds);

FD_SET(fd, &masterfds);
```

Şimdi **masterfds**'den **readfds** ye sonsuz döngü ilk kopyaları boyunca gidiyoruz, sonrasında **readfds** üzerinden **select()** çağrılır ve doğru olayın performansı (sokete **stdin** yollamak,yada soketten **stdout**'a giriş yapılır.) **readfds select()**'in hangi bitlerinin set olduğuna bağlıdır.Şuna dikkat etmek gerekir ki **EOF** ya **stdin** dendir yada soket sonlandırmalı telnet oturumundandır.

```
while (1) {  
    memcpy(&readfds, &masterfds, sizeof(fd_set));  
  
    if (select(fd+1, &readfds, NULL, NULL, NULL) < 0) {  
        perror("on select");  
        exit(1);  
    }
```

```
    if (FD_ISSET(0, &readfds)) {  
        n = read(0, buf, 100);  
        if (n == 0) {  
            close(0);  
            close(fd);  
            exit(1);  
        } else {  
            write(fd, buf, n);  
        }  
    }
```

```
    if (FD_ISSET(fd, &readfds)) {  
        n = read(fd, buf, 100);  
        if (n == 0) {  
            close(0);
```

```
    close(fd);  
    exit(1);  
} else {  
    write(1, buf, n);  
}  
}  
  
}  
}
```

Telnet iyi bir programdır çünkü müşteriler tarafından yaygın kullanılır. Net üzerinde var olan pek çok sunucu programı telnet üzerine bağlantı kurmak isteyen kullanıcılara bağlantı sunarlar. Örnek olarak **blackbird.wunderground** adresinden yayın yapan ve portu 3000 olan Michigan hava sunucusu gösterilebilir. Eğer bunu telnetlerseniz (yada **jteln**) şuan ki hava durumuna ulaşabilirsiniz. Bu çok güçlü bir paradigmadır. Telnet iyidir çünkü Michigan halkına müşteri kodu olmadan sunucunun kullanılmasına izin verir. Biz zaten müşteriye sahibiz: **Telnet**. Diğer bir örnek olarak da genel müşteri programları Netscape ve ftp gösterilir.