

# Sare zerbitzuak eta aplikazioak

## 2. Socketak

Nola komunikatu programak  
sarearen bitartez?

Socket interfazea

Socketa: komunikazio atea

C lengoaia

Socket motak

# Nola adierazi zein socket erabili

- Guk sortua:
  - Deskriptore bat (int).
- Kanpokoak:
  - Socket helbide bat (struct sockaddr).

# IPv4 socket baten helbidea

- *struct sockaddr\_in* egitura:
  - sin\_family = AF\_INET
  - sin\_port
  - sin\_addr
- *struct in\_addr* egitura:
  - s\_addr



# Helbideak: funtzio laguntzaileak

- Byte ordenazioa
  - “network to host” eta “host to network”.
- Puntuzko notazioa
  - “ascii to network” eta “network to ascii”.

Sare aplikazio adibidea

daytime

- UDP

- Zerbitzaria
- Bezeroa

- TCP

- Zerbitzaria
- Bezeroa

# Oinarrizko UDP zerbitzaria

- Socketa sortu.
- Socketari helbide bat esleitu.
- Begizta infinitua:
  - Komunikatu socketaren bitartez.

# Daytime zerbitzaria UDP erabiliz

```
int sock;
struct sockaddr_in zerb_helb, bez_helb;
socklen_t helb_tam;
char buf[MAX_BUF];
time_t ticks;

sock = socket(AF_INET, SOCK_DGRAM, 0);

zerb_helb.sin_family = AF_INET;
zerb_helb.sin_addr.s_addr = htonl(INADDR_ANY);
zerb_helb.sin_port = htons(PORT);

bind(sock, (struct sockaddr *) &zerb_helb,
    ↪ sizeof(zerb_helb));
```

# Daytime zerbitzaria UDP erabiliz

```
while(1)
{
    helb_tam = sizeof(bez_helb);
    recvfrom(sock, buf, MAX_BUF, 0,
        ↳ (struct sockaddr *) &bez_helb, &helb_tam);

    ticks = time(NULL);
    snprintf(buf, MAX_BUF, "%s", ctime(&ticks));
    sendto(sock, buf, strlen(buf), 0,
        ↳ (struct sockaddr *) &bez_helb, helb_tam);
}
```

# Oinarrizko UDP bezeroa

- Socketa sortu.
- Komunikatu socketaren bitartez.
- Socketa itxi.

# Daytime bezeroa UDP erabiliz

```
int sock, n;  
char buf[MAX_BUF];  
struct sockaddr_in zerb_helb;  
  
sock = socket(AF_INET, SOCK_DGRAM, 0);  
  
zerb_helb.sin_family = AF_INET;  
zerb_helb.sin_port = htons(PORT);  
inet_aton(argv[1], &zerb_helb.sin_addr);  
  
sendto(sock, "", 0, 0, (struct sockaddr *)  
↳ &zerb_helb, sizeof(zerb_helb));
```



# Daytime bezeroa UDP erabiliz

```
n = recvfrom(sock, buf, MAX_BUF, 0, NULL, NULL);  
buf[n] = 0;  
printf("Zerbitzaritik jasotako data: %s", buf);  
close(sock);
```

# Oinarrizko TCP zerbitzaria

- Socketa sortu.
- Socketari helbide bat esleitu.
- Socketa entzute socket gisa ezarri.
- Begizta infinitua:
  - Blokeatu konexio eskaera baten zain (elkarrizketa socketa).
  - Komunikatu elkarrizketa socketaren bitartez.
  - Itxi elkarrizketa socketa.

# Daytime zerbitzaria TCP erabiliz

```
int sock, elkarriketa;  
struct sockaddr_in zerb_helb;  
char buf[MAX_BUF];  
time_t ticks;  
  
sock = socket(AF_INET, SOCK_STREAM, 0);  
  
zerb_helb.sin_family = AF_INET;  
zerb_helb.sin_addr.s_addr = htonl(INADDR_ANY);  
zerb_helb.sin_port = htons(PORT);  
  
bind(sock, (struct sockaddr *) &zerb_helb,  
    ↪ sizeof(zerb_helb));
```

# Daytime zerbitzaria TCP erabiliz

```
listen(sock, 5);
```

```
while(1)
```

```
{
```

```
    elkarrizketa = accept(sock, NULL, NULL);
```

```
    ticks = time(NULL);
```

```
    snprintf(buf, MAX_BUF, "%s", ctime(&ticks));
```

```
    write(elkarrizketa, buf, strlen(buf));
```

```
    close(elkarrizketa);
```

```
}
```

# Oinarrizko TCP bezeroa

- Socketa sortu.
- Konektatu socketa beste socket batekin.
- Komunikatu socketaren bitartez.
- Socketa itxi.

# Daytime bezeroa TCP erabiliz

```
int sock, n;  
char buf[MAX_BUF];  
struct sockaddr_in zerb_helb;  
  
sock = socket(AF_INET, SOCK_STREAM, 0);  
  
zerb_helb.sin_family = AF_INET;  
zerb_helb.sin_port = htons(PORT);  
inet_aton(argv[1], &zerb_helb.sin_addr);  
  
connect(sock, (struct sockaddr *) &zerb_helb,  
↳ sizeof(zerb_helb));
```

# Daytime bezeroa TCP erabiliz

```
printf("%s zerbitzaritik jasotako data eta  
↳ ordua: ", argv[1]);  
  
while((n = read(sock, buf, MAX_BUF)) > 0)  
{  
    buf[n]=0;  
    printf("%s", buf);  
}  
close(sock);
```