

### Pregunta 1

```
/* Type to represent a port. */
typedef uint16_t in_port_t;

/* Structure describing an Internet socket address. */
struct sockaddr_in
{
    __SOCKADDR_COMMON (sin_);
    in_port_t sin_port;           /* Port number. */
    struct in_addr sin_addr;      /* Internet address.

    /* Pad to size of 'struct sockaddr'. */
    unsigned char sin_zero[sizeof (struct sockaddr) -
```

- El número de puerto

### Pregunta 2

- Dominio y tipo del socket
  - AF\_INET y SOCK\_DGRAM
- Uso de bzero
  - Escribe 0's en la dirección de memoria con n zeros. En este caso especial se deja la dirección del servidor en 0's
- Bits variable puerto
  - 4bytes -> 32 bits para puerto
  - 2bytes -> 16 bits para server\_addr.sin\_port
- Imprimir el valor de puerto antes y después de htons

```
eric@DESKTOP-NCQNCN8:/mnt/c/Users/etrejot1300/Desktop/PSL_practicas$ gcc programa9-1.c -o P9
eric@DESKTOP-NCQNCN8:/mnt/c/Users/etrejot1300/Desktop/PSL_practicas$ ./P9
1c20
201c
```

- ¿Que realiza htons?
  - Cambia el orden de los bytes de acuerdo a Little Endian.

### Pregunta 3

Se debe inicializa la variable server\_addr, que contendrá la dirección IP, el puerto y el dominio de comunicación del servidor.

La otra variable, msg\_to\_client\_addr, es una estructura para recibir la información de comunicación del cliente, que de momento contendrá basura.

### Pregunta 4

Para el cliente:

- **Dirección IP cliente:** msg\_to\_client\_addr.sin\_addr.s\_addr
- **Puerto cliente:** msg\_to\_client\_addr.sin\_port

Se inicializa server\_addr para:

- Asignar una **dirección IP** al servidor con la variable server\_addr.sin\_addr.s\_addr
- Asignar un **puerto** al servidor con la variable server\_addr.sin\_port y la función htons()
- Especificarle el **dominio de comunicación** internet AF\_INET al servidor con la variable server\_addr.sin\_family