



Instituto Politécnico Nacional

Escuela Superior de Cómputo



Desarrollo de Sistemas Distribuidos

“Capítulo 9: Servidor de Dominio Internet
Tipo UDP”

Profesor: Coronilla Contreras Ukranio

Grupo: 4CM4

Equipo 6:

- Grimaldo Peralta Marco Antonio
- Huerta Cortés Alan Antonio
- Macías Castillo Josue
- Vilches Segundo Galilea Yanely

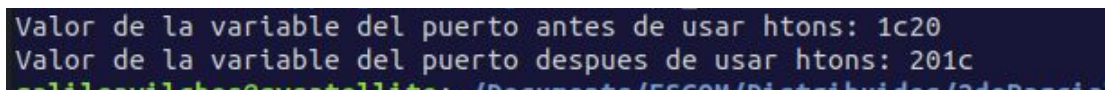
RESPUESTAS CAPÍTULO 9: SERVIDOR DE DOMINIO INTERNET TIPO UDP

PREGUNTA 9.1

1. *sin_port* contiene el número de puerto en orden de bytes de la red (*Big - endian*).
-

PREGUNTA 9.2

1. El dominio es Internet (AF_INET) y su tipo es Datagrama (SOCK_DGRAM).
2. La función *bzero* se encarga para llenar toda la estructura con ceros.
3. La variable *puerto* tiene un tamaño de 13 bits. Esto se logra simplemente haciendo la conversión del dato en cuestión (7200) a binario. Como resultado de dicha operación, obtenemos el número 1110000100000, que se compone de 13 bits.
4. Valor de la variable del puerto antes de usar *htons* en hexadecimal: 1c20 (7200)
Valor de la variable del puerto después de usar *htons* en hexadecimal: 201c (8220)



```
Valor de la variable del puerto antes de usar htons: 1c20
Valor de la variable del puerto despues de usar htons: 201c
galileos@ubuntu:~/Documents/55604/Distribuidos (2da Parcial)
```

Imagen 1 - Resultados obtenidos en la prueba 4

Lo que sucedió fue lo siguiente:

El número decimal 7200 se representa como 0x1c20. Sin embargo, muchas computadoras (incluida en la que se realizó la prueba) almacenan datos en formato *Little - endian*, haciendo que el 7200 se almacene de esta misma manera (0x1c20).

Dado que la función *htons* realiza una conversión a *Big - endian*, esto quiere decir que los bytes se intercambiarán si es necesario, poniendo el más

significativo al inicio. Es por esto que se imprime un valor 0x201c, que en decimal es 8220.

Cabe recalcar que, si la computadora trabaja desde un inicio en un formato *Big-endian*, la función *htons* ya no haría el intercambio.

PREGUNTA 9.3

1. Se inicializa la variable *server_addr* con ceros. Esto es gracias a la función *bzero* en la siguiente línea de código:
`bzero((char *)&server_addr, sizeof(server_addr));`
 2. Si la otra variable *msg_to_client_addr* no se inicializa, entonces los valores que deben de tener sus miembros son *valores basura*.
-

PREGUNTA 9.4

1. La dirección IP del cliente que haya realizado alguna petición al servidor se almacenaría en la estructura *msg_to_client_addr*, tal y como se especifica en la *Línea 4* del programa:
`recvfrom(s, (char *)num, 2*sizeof(int), 0, (struct sockaddr *)&msg_to_client_addr, &clilen);`
2. El mejor propósito que puede tener inicializar los miembros de la estructura *server_addr* es evitar que los datos sean erróneos, no logrando establecer bien los valores que se desean.