Examen Parcial ACS.
Edgar Daniel Barcenas Martinez

1.-Explica los 2 códigos DETALLADAMENTE, sobre todo enfatizando en el servidor los siguientes bloques:

```
void sigchld_handler(int s){
  while(wait(NULL) > 0);
}
```

¿Qué hace el wait(NULL)?

Esa función se utiliza como manejador para controlar los signals de un proceso hijo.Si algún proceso tiene más de un proceso hijo, luego de llamar a wait (), el proceso padre tiene que estar en estado de espera si ningún hijo termina.

Si solo se finaliza un proceso secundario, entonces devolver un wait () devuelve el ID del proceso del proceso secundario finalizado.

Si se finalizan más de un proceso hijo, entonces wait () cosecha cualquier hijo hijo arbitrariamente y devuelve un ID de proceso de ese proceso hijo.

Cuando wait () regresa, también definen el estado de salida (que le dice a nuestro, un proceso por qué terminó) a través del puntero, si el estado no es NULL .

Si algún proceso no tiene un proceso hijo, wait () devuelve inmediatamente "-1".

```
if (setsockopt(sockfd, SOL_SOCKET, SO_REUSEADDR, &yes, sizeof(int)) == -1){
  perror("Server-setsockopt() error lol!");
  exit(1);
}
else
  printf("Server-setsockopt is OK...\n");
```

¿Qué hace la función setsockopt(…)?

setsockopt () manipula opciones para el socket al que hace referencia el descriptor de archivo sockfd . Las opciones pueden existir en múltiples niveles de protocolo; siempre están presentes en el nivel superior del zócalo.

En caso de éxito, se devuelve cero. En caso de error, se devuelve -1 y se establece errno de manera adecuada.

my_addr.sin_addr.s_addr = INADDR_ANY;

¿Qué hace la macro INADDR_ANY?

Al usar INADDR_ANY

Admitido datos que provengan de cualquiera de los interfaces de red que tenga un servidor tenga y aunque el servidor cambie su IP, el programa seguirá funcionando sin cambios.

Examen Parcial ACS.
Edgar Daniel Barcenas Martinez

Explica este bloque (AQUI EL COMENTARIO LES DA UNA IDEA DE LO QUE HACE EL BLOQUE)

```
/* clean all the dead processes */
sa.sa_handler = sigchld_handler;
sigemptyset(&sa.sa_mask);
sa.sa_flags = SA_RESTART;
if(sigaction(SIGCHLD, &sa, NULL) == -1){
  perror("Server-sigaction() error");
  exit(1);
}
else
  printf("Server-sigaction() is OK...\n");
```

Esta parte del codigo limpia los procesos muertos, con sa.sa_handler = sigchld_handler Sirve para matar procesos zombie Usamos signation para hacer una llamada al sistema y con sa_handler para hacer la union con nuestra funcion que es la encargada de hacer esperar al padre hasta que termine el hijo Especificamos las señales que deben bloquearse y con la bandera restart indicamos que se pueda reiniciar el proceso A la funcion para verificar que no hay mas procesos mandamos sigchld donde obtine el id del proceso y la direccion de la estructura si devuelve -1 hay un error

Modifica levemente tanto cliente como servidor para que el mensaje a enviar sea capturado desde teclado, es decir, un chat en modo texto:

Examen Parcial ACS.
Edgar Daniel Barcenas Martinez