

Examen-Junio-2021-resuelto.pdf



Yassine_bhk



Fundamentos de Software de Comunicaciones



2º Grado en Ingeniería de las Tecnologías de Telecomunicación



Escuela Técnica Superior de Ingeniería de Telecomunicación
Universidad de Málaga

WUOLAH + BBVA

Llévate

15€

Al abrir tu Cuenta Online Sin Comisiones y hacer una compra superior a 15€.

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

Cuéntame más



¿Cómo?



ESTE CURSO NO CARGUES CON MÁS PESO
QUE EL DE TU CONCIENCIA POR APROBAR,
ÉL PESA MENOS DE 1 KG .

MSI™

```
/* CLIENTE */
```

```
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/sysinfo.h>
#include <signal.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
```

```
ssize_t readn (int fd, void * buffer, size_t n) {
```

```
    char * p = (char *) buffer;
    int leidos;
    int total_leidos = 0;
    int intentando_leer = n;
```

```
    do {
```

```
        errno = 0;
        leidos = read (fd, p + total_leidos, intentando_leer);
```

```
        if (leidos > 0) {
```

```
            intentando_leer -= leidos;
            total_leidos += leidos;
        }
```

```
    } while (total_leidos < n && leidos > 0 || errno == EINTR);
```

```
    if (leidos < 0) {
```

Rendimiento extraordinario, gran portabilidad y diseño compacto, la serie Prestige está diseñada para satisfacer todas las expectativas de un portátil perfecto para los negocios y la productividad. Certificado por la plataforma Intel® Evo™, el Prestige 13 Evo hace tu vida más productiva proporcionando experiencias extraordinarias en cada momento de tu trabajo y de tu vida.



MSI PRESTIGE 13

WUOLAH

```

        return leidos;
    }

    else {

        return total_leidos;
    }

}

ssize_t writen (int fd, void * buffer, size_t n) {

    char * p = (char *) buffer;
    int leidos;
    int total_leidos = 0;
    int intentando_leer = n;

    do {

        errno = 0;
        leidos = write (fd, p + total_leidos, intentando_leer);

        if (leidos > 0) {

            intentando_leer -= leidos;
            total_leidos += leidos;
        }

    } while (total_leidos < n && leidos > 0 || errno == EINTR);

    if (leidos < 0) {

        return leidos;
    }

    else {

        return total_leidos;
    }
}

```


GANAR 2 MALETAS LLENAS DE **Kinder** *bueno*

1. SUBE UNA FOTO
A @ CON TU
KINDER BUENO FAVORITO

2. DINOS CON QUE AMIGO
BUENO TE IRÍAS DE VIAJE

3. ¡ETIQUÉTANOS
Y DESCUBRE
SI HAS GANADO!



DESCUBRE MÁS EN @KINDERBUENO_ES

Promoción válida en España del 1/05/2023 al 30/06/2023 para mayores de 18 años.
Más información y Bases Legales en nuestro perfil de Instagram. @kinderbueno_es

```

    }

uint32_t resumen (char *b, int longitud) {

    uint32_t sum = 0;
    for (int i = 0; i < longitud; i++) {

        sum += b[i];
    }

    return sum*longitud;
}

int main (int argc, char * argv []) {

    if (argc < 4) {

        printf ("Porfavor, escriba un argumento");
        exit (1);
    }

    int sd = socket (AF_INET, SOCK_STREAM, 0);

    if (sd < 0) {

        perror ("Socket");
        exit (1);
    }

    struct sockaddr_in vinculo;
    vinculo.sin_addr.s_addr = inet_addr (argv [2]);
    vinculo.sin_family = PF_INET;
    vinculo.sin_port = htons (atoi (argv [1]));

    if (connect (sd, (struct sockaddr *) &vinculo, sizeof (vinculo)) < 0) {

        perror ("connect");
        close (sd);
        exit (1);
    }

    uint8_t nivel_seguridad;

```

```

int leidos = readn (sd, &nivel_seguridad, 1);
printf ("Nivel de seguridad: %d\n", nivel_seguridad);

    if (leidos < 0) {

        perror ("leyendo segur");
        close (sd);
        exit (1);
    }

int fd = open (argv [3], O_RDONLY);

    if (fd < 0) {

        perror ("open");
        close (sd);
        exit (1);
    }

uint16_t longitud;
uint16_t tipo;
uint16_t tipo_n;
uint16_t longitud_n;

int escritos;

do {

    char nombre [512];

    leidos = read (fd, nombre, 511);

        if (leidos < 0) {

            perror ("Leyendo datos");
            close (sd);
            exit (1);
        }

    nombre [leidos -1] = '\0';

    tipo = 1;
    tipo_n = htons (tipo);

```

ESTE CURSO NO CARGUES CON MÁS PESO
QUE EL DE TU CONCIENCIA POR APROBAR,
ÉL PESA MENOS DE 1 KG .

MSI™

Rendimiento extraordinario, gran portabilidad y diseño compacto, la serie Prestige está diseñada para satisfacer todas las expectativas de un portátil perfecto para los negocios y la productividad. Certificado por la plataforma Intel® Evo™, el Prestige 13 Evo hace tu vida más productiva proporcionando experiencias extraordinarias en cada momento de tu trabajo y de tu vida.



MSI PRESTIGE 13

```
longitud = strlen (nombre);
longitud_n = htons (longitud);

printf ("Tipo_n: %d\n", tipo_n);
printf ("longitud_n: %d\n", longitud_n);

printf ("Tipo: %d\n", tipo);
printf ("longitud: %d\n", longitud);
printf ("nombre: %s\n", nombre);

escritos = writen (sd, &tipo_n, 2);
escritos = writen (sd, &longitud_n, 2);
escritos = writen (sd, nombre, longitud);

if (escritos < 0) {

    perror ("escribiedno datos");
    close (sd);
    exit (1);
}

if (nivel_seguridad == 1) {

    uint32_t envio = resumen (nombre, longitud);
    uint32_t recibo_n;
    int l = readn (sd, &recibo_n, 4);
    uint32_t recibo = ntohs (recibo_n);

    if (recibo != envio) {

        printf ("Valor incorrecto de la función resumen. Posible
interceptación de los datos\n");
        break;
    }

}

} while (leidos > 0);

close (sd);

return 0;
```

WUOLAH

}


```

/* SERVIDOR */

#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/sysinfo.h>
#include <signal.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

#define PUERTO 4951

ssize_t readn (int fd, void * buffer, size_t n) {

    char * p = (char *) buffer;
    int leidos;
    int total_leidos = 0;
    int intentando_leer = n;

    do {

        errno = 0;
        leidos = read (fd, p + total_leidos, intentando_leer);

        if (leidos > 0) {

            intentando_leer -= leidos;
            total_leidos += leidos;
        }

    } while (total_leidos < n && leidos > 0 || errno == EINTR);
}

```

```

        if (leidos < 0) {

            return leidos;
        }

        else {

            return total_leidos;
        }

    }

ssize_t writen (int fd, void * buffer, size_t n) {

    char * p = (char *) buffer;
    int leidos;
    int total_leidos = 0;
    int intentando_leer = n;

    do {

        errno = 0;
        leidos = write (fd, p + total_leidos, intentando_leer);

        if (leidos > 0) {

            intentando_leer -= leidos;
            total_leidos += leidos;
        }

    } while (total_leidos < n && leidos > 0 || errno == EINTR);

    if (leidos < 0) {

        return leidos;
    }

    else {

        return total_leidos;
    }
}

```

ESTE CURSO NO CARGUES CON MÁS PESO
QUE EL DE TU CONCIENCIA POR APROBAR,
ÉL PESA MENOS DE 1 KG .

MSI™

Rendimiento extraordinario, gran portabilidad y diseño compacto, la serie Prestige está diseñada para satisfacer todas las expectativas de un portátil perfecto para los negocios y la productividad. Certificado por la plataforma Intel® Evo™, el Prestige 13 Evo hace tu vida más productiva proporcionando experiencias extraordinarias en cada momento de tu trabajo y de tu vida.



MSI PRESTIGE 13

```
    }

    }

    int fin = 0;

    void manejadora_sigint (int s) {

        fin = 1;
    }

    uint32_t resumen (char *b, int longitud) {

        uint32_t sum = 0;
        for (int i = 0; i < longitud; i++) {

            sum += b[i];
        }

        return sum*longitud;
    }

    int main (int argc, char * argv []) {

        if (argc < 2) {

            printf ("Porfavor, escriba un argumento");
            exit (1);
        }

        uint8_t nivel_seguridad = atoi (argv [1]);

        int sd = socket (AF_INET, SOCK_STREAM, 0);

        if (sd < 0) {

            perror ("Socket");
            exit (1);
        }

        struct sockaddr_in vinculo;
```

WUOLAH

```

vinculo.sin_addr.s_addr = INADDR_ANY;
vinculo.sin_family = PF_INET;
vinculo.sin_port = htons (PUERTO);

if (bind (sd, (struct sockaddr *) &vinculo, sizeof (vinculo)) < 0) {

    perror ("Vinculo");
    close (sd);
    exit (1);
}

if (listen (sd, 10) < 0) {

    perror ("Vinculo");
    close (sd);
    exit (1);
}

struct sockaddr_in info_cliente;
socklen_t tam = sizeof (info_cliente);

while (!fin) {

    int csd = accept (sd, (struct sockaddr *) &info_cliente, &tam);

    if (csd < 0) {

        perror ("csd");
        close (sd);
        exit (1);
    }

    int escritos = writen (csd, &nivel_seguridad, 1);

    if (escritos < 0) {

        perror ("escribiendo niv.seguridad");
        close (csd);
        close (sd);
        exit (1);
    }

    uint16_t tipo_n, longitud_n;

```

```

uint16_t tipo, longitud;

char nombre [512];
int leidos;

do {

    leidos = readn (csd, &tipo_n, 2);
    leidos = readn (csd, &longitud_n, 2);

    tipo = ntohs (tipo_n);
    longitud = ntohs (longitud_n);

    leidos = readn (csd, nombre, longitud);

    if (leidos < 0) {

        perror ("Leyendo datos");
        close (csd);
        close (sd);
        exit (1);
    }

    nombre [leidos] = '\0';

    escritos = writen (1, &tipo, 2);
    escritos = writen (1, &longitud, 2);
    escritos = writen (1, nombre, longitud);

    if (nivel_seguridad == 1) {

        uint32_t envio = resumen (nombre, longitud);
        escritos = writen (csd, &envio, 4);

        if (escritos < 0) {

            perror ("Leyendo datos");
            close (csd);
            close (sd);
            exit (1);
        }
    }
}

```



```
    } while (leidos > 0);  
  
    close (csd);  
  
}  
  
close (sd);  
  
return 0;  
}
```

ESTE CURSO NO CARGUES CON MÁS PESO
QUE EL DE TU CONCIENCIA POR APROBAR,
ÉL PESA MENOS DE 1 KG .

MSI™

/* EJERCICIO 1 */

```
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/sysinfo.h>
#include <signal.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
```

```
#define EST_OCIOSO 1
#define EST_ESP_ENV 2
#define EST_ENVIADO 3
```

```
#define EV_DATA 1
#define EV_SIGALARM 2
#define EV_SIGUSR1 3
#define EV_SIGUSR2 4
```

```
int fin = 0;
int pipe_fd [2];
int fifo_fd;
```

```
ssize_t writen (int fd, void * buffer, size_t n) {
```

```
    char * p = (char *) buffer;
    int escritos;
    int intentado_escribir = n;
    int total_escritos = 0;
```

```
    do {
```

```
        errno = 0;
        escritos = write (fd, p + total_escritos, intentado_escribir);
```

Rendimiento extraordinario, gran portabilidad y diseño compacto, la serie Prestige está diseñada para satisfacer todas las expectativas de un portátil perfecto para los negocios y la productividad. Certificado por la plataforma Intel® Evo™, el Prestige 13 Evo hace tu vida más productiva proporcionando experiencias extraordinarias en cada momento de tu trabajo y de tu vida.



MSI PRESTIGE 13

WUOLAH

```

        if (escritos > 0) {

            total_escritos += escritos;
            intentado_escribir -= escritos;
        }

    } while (escritos > 0 && total_escritos < n || errno == EINTR);

    if (escritos < 0) {

        return escritos;
    }

    else {

        return total_escritos;
    }
}

ssize_t readn (int fd, void * buffer, size_t n) {

    char * p = (char *) buffer;
    int escritos;
    int intentado_escribir = n;
    int total_escritos = 0;

    do {

        errno = 0;
        escritos = read (fd, p + total_escritos, intentado_escribir);

        if (escritos > 0) {

            total_escritos += escritos;
            intentado_escribir -= escritos;
        }

    } while (escritos > 0 && total_escritos < n || errno == EINTR);

    if (escritos < 0) {

        return escritos;
    }
}

```

```

    }

    else {

        return total_escritos;
    }
}

void manejadora_alarm (int s) {

    int evento = EV_SIGNAL;
    int escritos = writen (pipe_fd [1], &evento, 4);

    signal (s, manejadora_alarm);

}

void manejadora_usr1 (int s) {

    int evento = EV_SIGNAL;
    int escritos = writen (pipe_fd [1], &evento, 4);

    signal (s, manejadora_usr1);

}

void manejadora_usr2 (int s) {

    int evento = EV_SIGNAL;
    int escritos = writen (pipe_fd [1], &evento, 4);
    signal (s, manejadora_usr2);

}

int maximo (int a, int b) {

    if (a > b) {

        return a;
    }

    else {

```

```

        return b;
    }
}

int espera_evento () {

    int evento;

    fd_set conjunto;
    FD_ZERO (&conjunto);
    FD_SET (fifo_fd, &conjunto);
    FD_SET (pipe_fd [0], &conjunto);

    int max_fd = maximo (pipe_fd [0], fifo_fd);

    int r = select (max_fd + 1, &conjunto, 0, 0, 0);

    if (select < 0) {

        if (errno == EINTR) {

            errno = 0;
        }

        else {

            perror ("Select");
            exit (1);
        }
    }

    if (FD_ISSET (pipe_fd [0], &conjunto)) {

        int leidos = readn (pipe_fd [0], &evento, 4);

        if (leidos < 0) {

            perror ("Leyendo\n");
            exit (1);
        }

        return evento;
    }
}

```




Consigue una **beca de Fundación hna** para finalizar tus estudios

¡ÚLTIMOS DÍAS PARA SOLICITARLAS!

4° de
Arquitectura
Técnica
(o Grado
equivalente)

Máster
habilitante
en
Arquitectura



CONECTA by hna



36 BECAS de hasta **1.000€**
¡No te quedes sin la tuya!

```
    }

    if (FD_ISSET (fifo_fd, &conjunto)) {

        char buffer [512];
        int leidos;

        do {

            leidos = read (fifo_fd, buffer, 512);

            if (leidos < 0) {

                if (errno == EINTR) {

                    errno = 0;
                }

                else {

                    perror ("Leyendo\n");
                    exit (1);

                }

            }

            buffer [leidos - 1] = '\0';

            if (strcmp (buffer, "data") == 0) {

                return EV_DATA;
            }

            else {

                printf ("Cadena no reconocida. Terminando...\n");
                exit (1);
                fin = 1;
            }

        } while (leidos > 0 && errno == EINTR);
```

WUOLAH

```

    }

}

int main () {

    signal (SIGALRM, manejadora_alarm);
    signal (SIGUSR1, manejadora_usr1);
    signal (SIGUSR2, manejadora_usr2);

    int r = pipe (pipe_fd);

    if (r < 0) {

        perror("pipe");
        exit (1);

    }

    fifo_fd = open ("fcs_fifo", O_RDONLY);

    if (fifo_fd < 0) {

        perror ("fifo");
        exit (1);

    }

    printf ("Pid de mi proceso: %d\n", getpid ());

    int estado = EST_OCIOSO;
    struct itimerval timer;
    struct timeval t1 = {1, 500000};
    struct timeval t2 = {0,0};

    while (!fin) {

        int evento = espera_evento ();

        if (evento < 0) {

```

```

        printf ("Error espera evento");
        exit (1);
    }

    switch (estado)
    {
    case EST_OCIOSO:

        if (evento == EV_DATA) {

            estado = EST_ESP_ENV;
            printf ("Cambiando a estado esperando enviar...\n");
        }

        else {

            printf ("Evento no esperado");
        }

        break;

    case EST_ESP_ENV:

        if (evento == EV_SIGUSR1) {

            printf ("datos enviados...\n");
            estado = EST_ENVIADO;

            timer.it_interval = t1;
            timer.it_value = t1;
            setitimer (ITIMER_REAL, &timer, 0);
        }

        break;

    case EST_ENVIADO:

        switch (evento)
        {

            case EV_SIGALARM:

```

```

        estado = EST_ESP_ENV;
        printf ("Timeout: volviendo al estado esperando evniar...\n");

        break;

case EV_SIGUSR2:

    estado = EST_OCIOSO;
    printf ("Pasando a estado ocioso...\n");
    timer.it_interval = t2;
    timer.it_value = t2;
    setitimer (ITIMER_REAL, &timer, 0);

    break;

default:

    printf ("Evento no esperado\n");
    exit (1);

    break;
}

break;

default:

    printf ("Estado no esprado");
    exit (1);
    break;
}

}

return 0;
}

```

WUOLAH

(a nosotros por suerte nos pasa)

Pero me voy a graduar.
Mañana mi diploma y título he de pagar

Llegó mi momento de despedirte
Tras años en los que has estado mi
lado.

Siempre me has ayudado
Cuando por exámenes me he
agobiado

Oh Wuolah wuolithah
Tu que eres tan bonita

si lees esto me debes un besito

WUOLAH

