

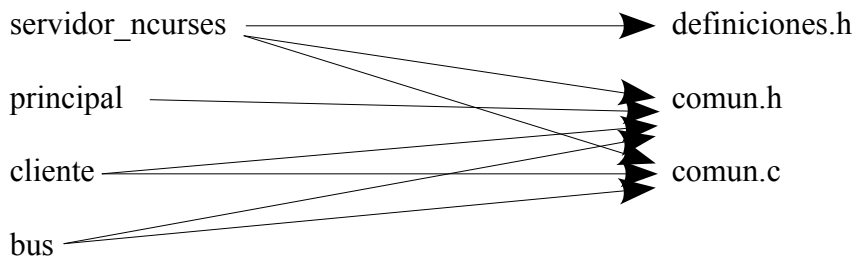
Índice

1. Ficheros.....	1
2. Esquema de creación.....	1
3. Esquema de sincronización.....	1
4. Esquema de comunicación.....	2
5. Funciones especiales.....	2
6. Procesos.....	5

1. Ficheros

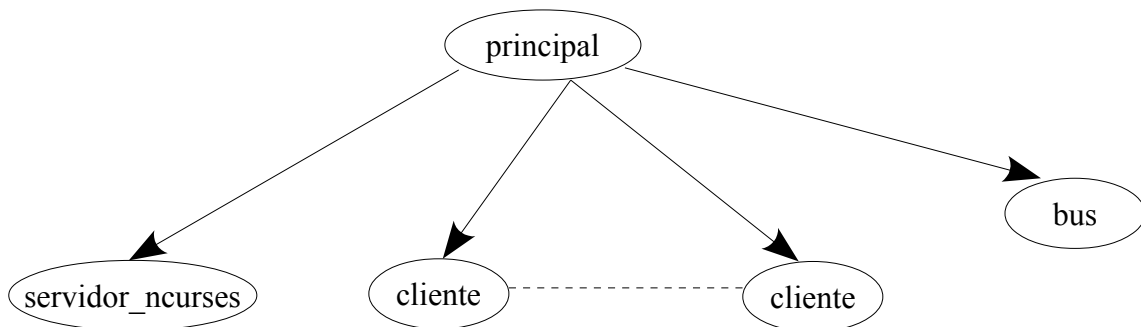
Procesos

Librerías



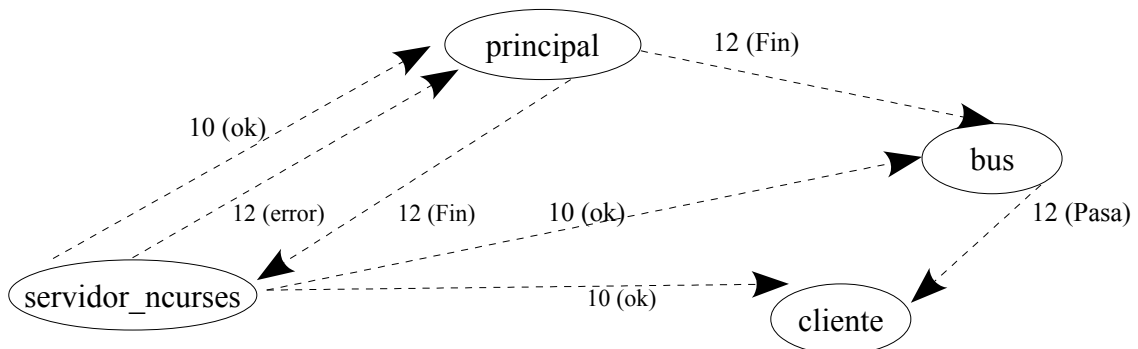
fichcola.txt → Para crear cola de mensajes

2. Esquema de creación

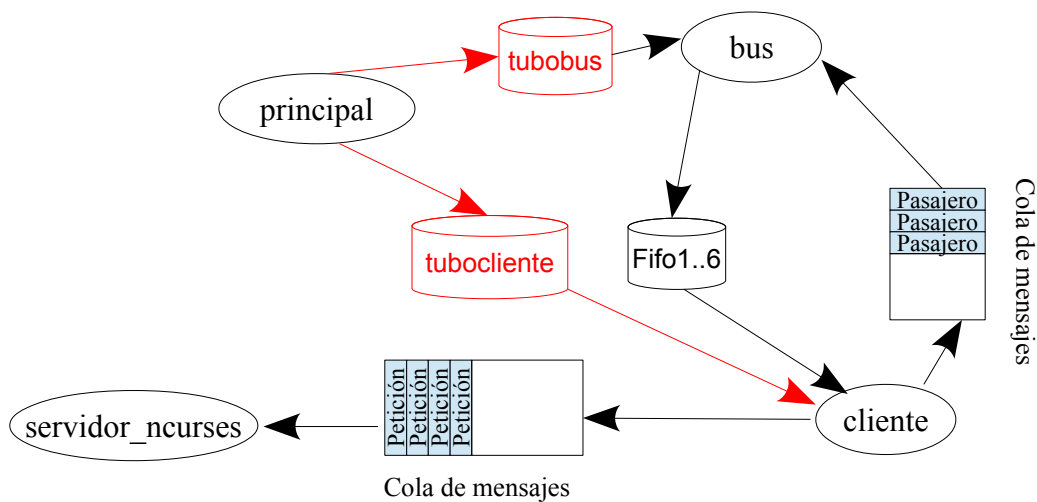


Bucle desde 1 hasta 30 (maxclientes)

3. Esquema de sincronización



4. Esquema de comunicación



5. Funciones especiales

Función leeparametros

(Va en principal.c)

```
void leeparametros (struct ParametrosBus *parambus, struct
ParametrosCliente *paramclientes, int *maxclientes, int
*creamin,int *creamax)
{
    int ok=0;

    *maxclientes=30; //Numero de clientes que se crearan
    *creamin=1;//Intervalo de tiempo para crear nuevos clientes MIN
    *creamax=5;//Intervalo de tiempo para crear nuevos clientes MAX
    parambus->numparadas=paramclientes->numparadas=6;//Cantidad
paradas
    parambus->capacidadbus=5; //Capacidad del bus
    parambus->tiempotrayecto=3;// Tiempo del trayecto entre paradas
    paramclientes->aburrimientomax=12;//Tiempo en aburrirse MAX
    paramclientes->aburrimientomin=20;//Tiempo en aburrirse MIN

    while(ok == 0)
    {
        system("clear");
        printf("Valores de los parámetros...\n\n");
        printf("Numero de pasajeros que se crearan:
%d\n", *maxclientes);
        printf("Intervalo de tiempo para crear nuevos pasajeros:
[%d-%d] \n", *creamin, *creamax);
        printf("Número de paradas: %d\n", parambus->numparadas);
        printf("Capacidad del Bus: %d\n", parambus->capacidadbus);
        printf("Tiempo en el trayecto entre paradas: %d\n", parambus-
>tiempotrayecto);
```

```

        printf("Intervalo de tiempo de aburrimiento: [%d-
%d]\n",paramclientes->aburrimientomin,paramclientes-
>aburrimientomax);
        printf("Pulse 0 si desea introducir nuevos valores,
cualquier otro valor si desea continuar.\n");
        scanf("%d",&ok);

        if(ok == 0){
            do{
                printf("Numero de pasajeros que se crearan [maximo
50]:\n");
                scanf("%d",maxclientes);
            }while(*maxclientes <= 0 || *maxclientes > 50);

            do{
                printf("Intervalo de tiempo para crear nuevos pasajeros
MIN [entre 1 y 8]: \n");
                scanf("%d",creamin);
            }while(*creamin< 1 ||*creamin > 8 );

            do{
                printf("Intervalo de tiempo para crear nuevos pasajeros
MAX [entre 2 y 20]: \n");
                scanf("%d",creamax);
            }while(*creamax<2 || *creamax>20 || *creamax<=*creamin);

            do{
                printf("N mero de paradas: \n");
                scanf("%d",&parambus->numparadas);
            }while(parambus->numparadas <2 || parambus->numparadas>
6);
            paramclientes->numparadas=parambus->numparadas;

            do{
                printf("Capacidad del bus [maximo 10]: \n");
                scanf("%d",&parambus->capacidadbus);
            }while(parambus->capacidadbus<=0 || parambus->capacidadbus
>10);

            do{
                printf("Tiempo en el trayecto entre paradas [maximo
10]:\n");
                scanf("%d",&parambus->tiempotrayecto);
            }while(parambus->tiempotrayecto < 1 ||parambus-
>tiempotrayecto > 10 );

            do{
                printf("Intervalo de tiempo en esperar para aburrirse
MIN [entre 10 y 20]:\n");
                scanf("%d",&paramclientes->aburrimientomin);
            }while(paramclientes->aburrimientomin<1 || paramclientes-
> aburrimientomin>10 );

```

```

        do{
            printf("Intervalo de tiempo en esperar para aburrirse
MAX [entre 15 y 40]:\n");
            scanf("%d",&paramclientes->aburrimientomax);
        }while(paramclientes->aburrimientomax < 5 ||
paramclientes->aburrimientomax > 20 || paramclientes-
>aburrimientomin > paramclientes->aburrimientomax);
    }
}
}

```

Función creaproceso

(Va en principal.c)

```

int creaproceso(const char nombre[],int tubo)
{

    int vpid;

    vpid=fork();
    if(vpid==0)
    {
        close(2);
        dup(tubo);
        execl(nombre,nombre,NULL);
        perror("error de execl");
        exit(-1);
    }
    else if (vpid== -1)
    {
        perror("error de fork");
        exit(-1);
    }
    return vpid;
}

```

6. Procesos

principal.c

```
main()
{
    Prepararse para recibir señales 10 y 12;
    Inicializamos semilla;
    Leemos los parámetros;
    Crear servidor_ncurses con creaservigraf; (*)
    Esperar señal de ok o de error del servidor gráfico;
    Creamos las fifos;      (*)
    Creamos tuberías;
    Creamos el bus; (*)
    Escribimos en tubería parámetros del bus;
    for (i=1;i<=maxclientes;i++)
    {
        Crear cliente con creaproceso; (*)
        Escribimos en tubería parámetros del cliente;
        Espera aleatoria antes de crear otro cliente; (*)
    }
    Espera fin de sus hijos; (*)
    Avisa al bus de fin con señal 12;
    Avisa a servidor gráfico con señal 12;
    Cerramos y borramos las fifos; (*)
}
```

cliente.c

```
main()
{
    Prepararse para recibir señal 10;
    Prepararse para recibir señal 12;
    Crear la cola de mensajes;
    Crear cola de mensajes entre bus y cliente;
    Inicializamos semilla;
    Leemos parámetros de la tubería;
    Generar parada de llegada y de bajada; (*)
    Abre fifo de la parada de bajada;
    Se visualiza en la parada de llegada;
    Escribe en la cola de mensajes entre bus y cliente;
    Espera llegada del bus (señal 12);
    Se borra;
    Se pinta en el bus;
    Espera testigo de parada de bajada;
    Se borra del bus;
    Se visualiza en la parada de bajada;
}
```

bus.c

```
main()
{
    Prepararse para recibir señal 10 y 12;
    Inicializamos semilla;
    Indicamos numero de asientos libres;
    Crear las dos colas de mensajes;
    Leemos parámetros de la tubería;
    Abrimos todas las fifos; (*)
    Inicializamos el numero de montados de cada parada; (*)
    while (1)
    {
        for (parada=1;parada<=params.numparadas;parada++)
        {
            Se pinta el bus en la parada;
            Bajamos a la gente de esa parada;
            Indicamos que no hay nadie montado en esa parada;
            Montamos a la gente de esa parada;
            Se pinta bus entre paradas; (*)
            Se espera un tiempo de recorrido; (*)
        }
    }
}
```

Abrimos todas las fifos;

```
for (i=1;i<=params.numparadas;i++)
{
    sprintf(nombrefifo,"fifo%d",i);
    fifos[i]=open(nombrefifo,O_WRONLY);
}
```

Inicializamos el numero de montados de cada parada;

```
for (i=1;i<=params.numparadas;i++) montados[i]=0;
```

Bajamos a la gente de esa parada;

```
for (gente=1;gente<=montados[parada];gente++)
{
    write(fifos[parada], &testigo, sizeof[testigo]);
    libres++;
    retardo;
}
```

Montamos a la gente de esa parada;

```
while (haya espacio en el bus && haya pasajeros esperando en
      esa parada)
{
    Avisamos al pasajero para que suba al bus;
    libres--;
    Aumentamos el numero de montados para la parada de
    bajada del pasajero que acaba de subir;
}
```

Se pinta bus entre paradas;

```
if (parada == params.numparadas) pinta (colagrafica,
parada*10+1);
else pinta (colagrafica, parada*10+parada+1);
```