LARC. Curso 2010/11.

NOTAS SOBRE LA IMPLEMENTACIÓN DE LA PRÁCTICA 3

COMPILACIÓN Y EJECUCIÓN DEL PROGRAMA.

Los ficheros funciones.h y funciones.c (disponibles en aula virtual) contienen la implementación de las funciones **actualizarbuffer** y **entregarbuffer** (VÉASE EN "funciones.h" EL PROTOTIPO DE AMBAS). Dichos ficheros deben ser copiados al directorio con el código de la práctica 3.

Cada grupo debe crear un fichero .c con su **main** e incluir el fichero "funciones.h" para la compilación. Tanto el nombre del fichero con el **main** como el nombre del ejecutable final será libre. Además de este fichero .c, cada grupo puede utilizar los ficheros .h y .c que considere oportunos para el desarrollo de la práctica.

El fichero "funciones.h" contiene una función adicional **inicializar()** que debe ser llamada una sola vez al comienzo del programa.

Dicho fichero declara además las macros NMAX y T, cuya misión se especifica en el guión de la práctica 3.

Así, la estructura típica del programa será:

```
#include "funciones.h"       //… y los include necesarios

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

// Declaración de variables

inicializar();

// Uso libre de las funciones actualizarbuffer, entregarbuffer
// y de las macros T y NMAX
}
```

La implementación realizada de **actualizarbuffer** realiza lo siguiente:

- Selecciona como destino el equipo ANTERIOR en el anillo
- Crea un buffer de tamaño bufferlen que es variable (dicho buffer TIENE 0s internedios, por tanto no debe tratarse como una cadena de caracteres)

EJECUCIÓN Y MENSAJES DE DEPURACIÓN

La función **entregarbuffer** comprueba internamente si los mensajes recibidos son correctos. En este caso imprime por pantalla un mensaje indicándolo. En caso contrario (longitud del paquete global incorrecta o datos erroneos) se imprime un mensaje indicando el motivo del error **Y FINALIZA LA EJECUCIÓN DEL PROGRAMA EN ESTE PUNTO!**

En caso de que un grupo desee utilizar mensajes de depuración propios en el código, debe introducirlos dentro del siguiente código de compilación condicional

```
#ifdef DEPURACION
        // AQUI LOS PRINTF
#endif
```

y activar la variable DEPURACION en el Makefile (incluyendo en CARGS –DDEPURACION). NO SE ADMITIRÁN PRÁCTICAS QUE VUELQUEN MENSAJES A PANTALLA SIN ESTA DIRECTIVA.

El programa debe ser ejecutado en cada nodo del anillo. Durante la ejecución cada nodo envía continuamente al nodo anterior mensajes. Se sugiere que realice las pruebas iniciales usando anillos con dos nodos (-DSIMPLERING). Para las pruebas globales se sugiere configurar un valor en la directiva PROBABILIDAD (definida en libring.h) de 0.9 para evitar excesivas pérdidas de mensajes.

NOTAS

Nótese que un nodo sólo necesita llevar una variable con el anterior número de secuencia para hacer el control de los duplicados ya que sólo recibe mensajes de un emisor, y éste sólo le envía a él.

La función **entregarbuffer** LIBERA AUTOMATICAMENTE el buffer que le es entregado, por tanto, no es necesario que se llame a **free** desde el código principal para liberar esa zona de memoria.

ENTREGA

La entrega se realizará con el procedimiento habitual, copiando en la carpeta entregas/practica3 de la cuenta de cada grupo. Deben depositarse los siguientes ficheros:

- Códigos .c y .h propios
- Mediafifo.h, Mediafifo.c, libring.h, libring, funciones.h, funciones.c
- Makefile