

## Objetivos

### Objetivo general

Resolver un problema de concurrencia empleando las **primitivas POSIX** específicas de paso de mensajes

### Objetivos específicos

El alumno deberá adquirir las habilidades para la resolución de un problema concurrente utilizando los mecanismos de paso de mensajes entre procesos.

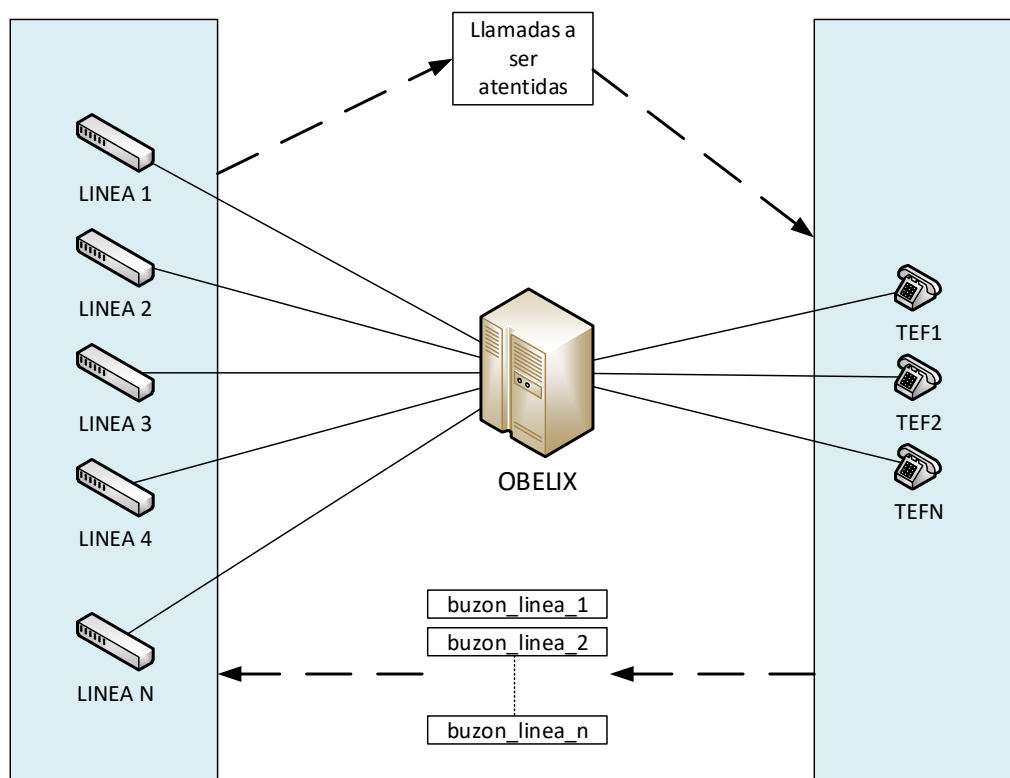
Implementar la solución teórica obtenida mediante los mecanismos que proporciona el estándar POSIX. Estos son los llamados IPC y más concretamente la implementación de paso de mensajes.

## Especificación

Estamos muy emocionados, y seguro que vosotros también, porque hemos descubierto la implementación de paso de mensajes entre procesos. Esto nos permite seguir trabajando sobre nuestro maravilloso sistema **Obelix**.

En este sistema, nos encontraremos con un proceso Manager que será el encargado del control de **Obelix**. Tendremos, por un lado, *líneas* que recibirán llamadas telefónicas, y, por otro lado, tendremos *teléfonos* que serán los encargados de atender estas llamadas.

Este sería el esquema de Obelix para paso de mensajes.



Construya un sistema compuesto por tres ejecutables que simule el funcionamiento que se detalla a continuación, y que es, la especificación básica de funcionamiento del sistema **Obelix**.

- **Manager:** Deberá cumplir las siguientes especificaciones:
  - Inicializar las colas de mensajes necesarias para establecer la comunicación entre líneas y teléfonos, así como, la comunicación entre los teléfonos y cada una de las líneas.
  - Lanzará el número de procesos *teléfono* y procesos *línea* definidos en este caso como constantes del sistema.
  - El sistema, gracias al paso de mensajes, no tendrá finalización, por lo que el *Apagado Controlado* de **Obelix** se realizará mediante la pulsación de Ctrl-C.
    - Forzará la finalización de todos los procesos *línea* actualmente en funcionamiento.
    - Forzará la finalización de todos los procesos *teléfono* existentes.
  - Por último, pero no menos importante, liberará todos los recursos utilizados. (tablas de procesos, colas de mensajes, etc.)
- **Teléfono:** Realizará las siguientes tareas dentro de su función:
  - Se inician en espera de una llamada, informando de ello: “Teléfono[UID] en espera...”
  - Cuando reciben una llamada, tienen una conversación variable entre 10..20 segundos e informan de ello: “Teléfono[UID] en conversación de llamada desde la línea: [buzon\_línea\_n]...”.
  - Cuando finalicen su conversación, notificarán a la línea este suceso y mostrarán un mensaje: Teléfono [%d] ha colgado la llamada. [buzon\_línea\_n]
  - Volverán a su estado inicial de espera de llamada.
  - *En ningún caso finalizarán su ejecución salvo cuando el proceso Manager lo considere oportuno.*
- **Línea:** Realizará las siguientes tareas dentro de su función:
  - Recibirá como parámetro su cola de mensajes para el proceso de notificación de teléfonos a líneas (buzon\_línea\_n)
  - Se inician y esperan una llamada (Simulado con un rand() entre 1..30 segundos), informando de ello: “Línea[UID] esperando llamada”.
  - Cuando reciben la llamada (Finalización de rand()):
    - Envió un mensaje a la *cola de llamadas para ser atendidas*.
    - Esperarán la notificación desde un teléfono, de que la llamada ha finalizado.
    - Notificarán la recepción del teléfono y volverán al estado de espera de llamada.
  - *En ningún caso finalizarán su ejecución salvo cuando el proceso Manager lo considere oportuno.*



## Ejemplo de ejecución

Una vez resuelto el ejercicio, si se ejecuta el proceso *manager*;

\$ exec/manager

el resultado debe ser similar al siguiente (cambiará el PID de los procesos, el orden de impresión y los valores generados de manera aleatoria):

jesus@asus:/ \$ exec/manager	Linea [49721] esperando fin de conversacion...
Linea [49717] esperando llamada...	Linea [49726] recibida llamada (/buzon_linea_9)...
Linea [49719] esperando llamada...	Linea [49726] esperando fin de conversacion...
[MANAGER] 10 lineas creadas.	Linea [49725] recibida llamada (/buzon_linea_8)...
Linea [49718] esperando llamada...	Linea [49725] esperando fin de conversacion...
Linea [49724] esperando llamada...	Teléfono [49727] ha colgado la llamada. /buzon_linea_2
Linea [49722] esperando llamada...	Teléfono [49727] en espera...
Linea [49725] esperando llamada...	Teléfono [49727] en conversacion de llamada desde Linea: /buzon_linea_7
Linea [49723] esperando llamada...	Linea [49719] conversación finalizada...
Linea [49726] esperando llamada...	Linea [49719] esperando llamada...
Linea [49721] esperando llamada...	Teléfono [49729] ha colgado la llamada. /buzon_linea_0
Linea [49720] esperando llamada...	Teléfono [49729] en espera...
[MANAGER] 3 teléfonos creados.	Teléfono [49729] en conversacion de llamada desde Linea: /buzon_linea_1
Teléfono [49728] en espera...	Linea [49722] recibida llamada (/buzon_linea_5)...
Teléfono [49727] en espera...	Linea [49722] esperando fin de conversacion...
Teléfono [49729] en espera...	Linea [49717] conversación finalizada...
Linea [49723] recibida llamada (/buzon_linea_6)...	Linea [49717] esperando llamada...
Linea [49723] esperando fin de conversacion...	Teléfono [49728] ha colgado la llamada. /buzon_linea_3
Teléfono [49728] en conversacion de llamada desde Linea: /buzon_linea_6	Teléfono [49728] en espera...
Teléfono [49728] ha colgado la llamada. /buzon_linea_6	Teléfono [49728] en conversacion de llamada desde Linea: /buzon_linea_6
Teléfono [49728] en espera...	Linea [49720] conversación finalizada...
Linea [49723] conversación finalizada...	Linea [49720] esperando llamada...
Linea [49723] esperando llamada...	Teléfono [49727] ha colgado la llamada. /buzon_linea_7
Linea [49719] recibida llamada (/buzon_linea_2)...	Teléfono [49727] en espera...
Linea [49719] esperando fin de conversacion...	Teléfono [49727] en conversacion de llamada desde Linea: /buzon_linea_4
Teléfono [49727] en conversacion de llamada desde Linea: /buzon_linea_2	Linea [49724] conversación finalizada...
Linea [49717] recibida llamada (/buzon_linea_0)...	Linea [49724] esperando llamada...
Linea [49717] esperando fin de conversacion...	Teléfono [49729] ha colgado la llamada. /buzon_linea_1
Teléfono [49729] en conversacion de llamada desde Linea: /buzon_linea_0	Teléfono [49729] en espera...
Linea [49720] recibida llamada (/buzon_linea_3)...	Teléfono [49729] en conversacion de llamada desde Linea: /buzon_linea_9
Linea [49720] esperando fin de conversacion...	Teléfono [49728] ha colgado la llamada. /buzon_linea_6
Teléfono [49728] en conversacion de llamada desde Linea: /buzon_linea_3	Teléfono [49728] en espera...
Linea [49724] recibida llamada (/buzon_linea_7)...	Teléfono [49728] en conversacion de llamada desde Linea: /buzon_linea_8
Linea [49724] esperando fin de conversacion...	Linea [49718] conversación finalizada...
Linea [49718] recibida llamada (/buzon_linea_1)...	Linea [49718] esperando llamada...
Linea [49718] esperando fin de conversacion...	Linea [49723] conversación finalizada...
Linea [49723] recibida llamada (/buzon_linea_6)...	Linea [49723] esperando llamada...
Linea [49723] esperando fin de conversacion...	^C
Linea [49721] recibida llamada (/buzon_linea_4)...	[MANAGER] Terminacion del programa (Ctrl + C).

----- [MANAGER] Terminar con cualquier proceso pendiente ejecutándose ----- - [MANAGER] Terminando proceso LINEA [49717]... [MANAGER] Terminando proceso LINEA [49718]... [MANAGER] Terminando proceso LINEA [49719]... [MANAGER] Terminando proceso LINEA [49720]... [MANAGER] Terminando proceso LINEA [49721]... [MANAGER] Terminando proceso LINEA [49722]... [MANAGER] Terminando proceso LINEA [49723]... [MANAGER] Terminando proceso LINEA [49724]... [MANAGER] Terminando proceso LINEA [49725]... [MANAGER] Terminando proceso LINEA [49726]... [MANAGER] Terminando proceso TELEFONO [49727]... [MANAGER] Terminando proceso TELEFONO [49728]... [MANAGER] Terminando proceso TELEFONO [49729]...	
--	--

## Sugerencias

- Crear antes los procesos de línea que los de teléfono. Si entra una llamada se quedará almacenada en la cola de “Llamadas a ser atendidas” hasta que estén disponibles los teléfonos.
- El sistema operativo tiene un máximo de 10 mensajes para las colas de mensajes, por lo que el número de líneas de Obelix será de 10.
- Ojo con las colas. Son independientes de los procesos y, por tanto, si la cola se queda creada y con mensajes, en la siguiente ejecución de los procesos nos pueden aparecer resultados no controlados.

## Bibliografía básica

- Kernighan B. W., Ritchie D. M. - *El lenguaje de programación C* - Ed. Prentice Hall 1991.
- Páginas del manual electrónico de UNIX.
- Rochkind, M.J. - *Advanced Unix Programming (2 nd Edition)* - Ed. Addison-Wesley 2004.
- Robbins, K.A., Robbins, S – *UNIX Programación Práctica* – Ed. Prentice Hall 1997.

## Plazos de entrega

Esta práctica deberá:

- **Entregarse como máximo el día 16 de mayo de 2023 a las 23:59 horas.**
- No se tendrán en cuenta las prácticas entregadas fuera de plazo.
- No se admitirán prácticas entregadas por email.
- La entrega de la práctica tendrá que realizarse en formato electrónico.
- La entrega será mediante una tarea en el espacio virtual de la asignatura en *Campus Virtual*. Si algún alumno no está registrado en dicha plataforma, deberá informar al profesor de prácticas para que le añada al curso.



- El fichero comprimido subido al sistema deberá cumplir la especificación que se indica en la sección Normas de Presentación.

## Normas de presentación

El seguimiento de las normas de presentación de la práctica es **obligatorio**, especialmente la parte relativa a la compilación automática mediante *make* de todos los ejecutables necesarios para la resolución de la misma. Obviamente, las rutas del fichero *makefile* deberán ser relativas para permitir su compilación en diferentes computadores sin necesidad de realizar ningún cambio. En este curso únicamente será posible entregar la práctica mediante *Campus Virtual*, es decir, **no hay que realizar ninguna impresión** de la misma.

Únicamente se subirá un fichero comprimido en formato zip por un solo miembro de la pareja. Debe cumplir el siguiente patrón de nombrado:

`primerApellido1_segundoApellido1_alumno1_primerApellido2_segundoApellido2_alumno2.zip`

Al descomprimir el fichero, se deberá obtener la siguiente estructura de directorios:

<b>/include</b>	Contendrá los ficheros de cabecera .h
<b>/src</b>	Contendrá los ficheros fuente .c
<b>/obj</b>	Contendrá los ficheros objeto (se deberán crear mediante make).
<b>/exec</b>	Contendrá los ejecutables (se deberán crear mediante make).
<b>/data</b>	Contendrá los ficheros para ejecutar las pruebas.
<b>/doc</b>	Contendrá un fichero .pdf con la memoria de la práctica.
<b>/</b>	El directorio raíz contendrá únicamente dos ficheros: <ul style="list-style-type: none"><li>• Un archivo makefile que deberá permitir compilar todos los programas relacionados con la práctica mediante la llamada directa a make.</li><li>• Un fichero ascii (author.txt) que contendrá nombre y apellidos de los autores.</li></ul>

La **memoria** de la práctica debe incluir una discusión de la solución planteada, haciendo hincapié en aquellas partes del código fuente que considere más relevante. Así mismo, la memoria ha de incluir una sección en la que se indique brevemente cómo compilar y ejecutar.

## Evaluación

Se valorarán dos aspectos: funcionalidad y calidad. La funcionalidad valora que se resuelva el problema conforme a sus especificaciones y que sea consistente, es decir, funcione adecuadamente en todas y cada una de las ejecuciones. Si solo funciona a veces, se considera que no funciona. Si existen casos no contemplados o faltan mecanismos de gestión de errores, la puntuación se verá afectada negativamente, aunque el programa nunca falle. En cuanto a la calidad del código, se valoran los siguientes aspectos:

- Estructura y organización del código, mediante clases, funciones o métodos.
- Elección de nombres significativos para funciones, variables, etc.
- Evitar uso de variables globales innecesarias.
- Evitar anidación injustificada de estructuras de control.
- Evitar uso incorrecto de tipos de datos, código redundante, código muerto, etc.
- Evitar bloques (funciones/métodos) demasiado largos.

La **valoración de la calidad es negativa**, es decir, restará sobre la nota final obtenida en funcionalidad. Esto implica que una práctica completamente funcional puede no obtener la máxima nota. Es importante destacar que el alumno debe ser capaz de defender su trabajo, por lo que en cualquier momento se le puede solicitar que defienda el trabajo presentado.

## Penalizaciones

Se contemplan 2 motivos por los que la puntuación total de la actividad puede sufrir penalizaciones:

- Errores de sintaxis que impiden ejecutar el programa.
- Incumplir otras condiciones del enunciado.

En el caso de errores de sintaxis, si supone cambios menores, se pedirá al alumno presentar una versión corregida.

## Condiciones

El alumno debe tener especial cuidado con la **incorporación de código ajeno** en su programa. Está PERMITIDO siempre que se cumplan las siguientes condiciones:

- Los fragmentos copiados y su autor están claramente identificados.
- El alumno comprende y es capaz de explicar con TODO DETALLE la funcionalidad del código copiado.
- El autor del código original permite la copia mediante la licencia correspondiente y esta aparece claramente junto al código original.
- El autor del original mantiene su código públicamente accesible (y el alumno proporciona el enlace en forma de comentario).
- El autor del original no es ni ha sido alumno de la UCLM.

**Infringir cualquiera de estas normas al utilizar código ajeno se considerará plagio. Cometer plagio implica una calificación de 0 en la actividad «Realización de prácticas de laboratorio»** según el artículo 9 de la «Guía de Evaluación del Estudiante»:

*Art. 9. Realización fraudulenta de pruebas de evaluación.*

*1. La constatación de la realización fraudulenta de una prueba de evaluación o el incumplimiento de las instrucciones fijadas para la realización de la prueba dará lugar a la calificación de suspenso (con calificación numérica de 0) en dicha prueba. En el caso particular de las pruebas finales, el suspenso se extenderá a la convocatoria correspondiente.*