



## Proyecto

### Fase II

#### Multiplexación y demultiplexación

#### Parte 1

### 1. Objetivos

#### Objetivo general:

Experimentar con multiplexación/demultiplexación de datos **limitada**, particularidades del sistema de archivos, métodos particulares de sincronización, y protocolos simples de comunicación.

#### Objetivos específicos:

- Comparar la instanciación por procesos y por hilos para paralelizar tareas
- Experimentar con sincronización mediante el uso de buzones
- Aplicar multiplexación y demultiplexación de datos
- Explorar aspectos de manejo de archivos de parte del Sistema Operativo
- Practicar con fragmentación y reensamblaje de datos, tanto a nivel de manejo de buffers, como de paquetización de datos para envío y recepción
- Explorar aspectos de administración de memoria en general, y de buffers específicos en particular

### 2. Descripción

Se desea transmitir un grupo de fotografías (formatos .jpeg, .png, o similares) entre dos máquinas físicamente separadas.

Existe un programa **lector** que recibe como argumento (por línea de comandos o por entrada estándar) la dirección del directorio donde se encuentran las fotos. Una vez ubicadas las fotos, su responsabilidad es subcontratar varios procesos para que lean las fotos. El **lector** bajo ninguna circunstancia debe leer el contenido de las fotos.

Un proceso **contratista** recibe un “puntero” (una “dirección”) hacia la foto que le toca procesar, y su responsabilidad es leerla (según las reglas que se indican más adelante), y pasársela a otro proceso local llamado el **emisor**. El proceso **contratista** es creado por el proceso **lector**, y una vez que se confirma la recepción exitosa de la foto, el hijo debe cerrarse.

A cada **contratista** se le encarga la transmisión de exactamente una foto. El **lector** debe abrir tantos **contratistas** como fotos haya que enviar, sin esperar a que un **contratista** haya concluido la transmisión para abrir otro.

Los procesos **contratistas** deben leer la foto en pedazos de a lo más 512 bytes. Las piezas deben ser particionadas aún más, en pedazos de a lo más 128 bytes. Los pedazos de 128 bytes deben ser enviados, usando **buzón de mensajes**, al proceso **emisor**. En la memoria de un proceso contratista nunca deben haber más de 1280 bytes de la foto.

**Restricción temporal:** Exactamente **dos contratistas** simultáneos.

El proceso **emisor** es independiente del proceso **lector** y es semi-permanente. Debe estar ya levantado antes de que el programa **lector** arranque.



El proceso **emisor** recibe pedacitos de foto de varios **contratistas**, y su responsabilidad es enviar estos pedacitos a su corresponsal, el proceso **receptor**, que se encuentra en otra máquina, por medio de una única conexión. Esto es, el **emisor** multiplexa las corrientes de pedacitos de los **contratistas** para enviarlas. Los pedacitos de diferentes fuentes deben ser combinados en esa única corriente. El **emisor** no debe esperar concluir con un **contratista** para comenzar con otro.

El proceso **receptor** debe recibir la corriente multiplexada de pedacitos de foto, y debe demultiplexarla para armar la fotos originales. El receptor debe manejar la demultiplexación por hilos, donde cada hilo se encarga de una foto específica. Ningún hilo debe tener más de 768 bytes activos de buffer en memoria.

**Restricciones:**

1. La comunicación entre el emisor y el receptor se debe hacer por medio de sockets TCP.
2. Deben usar C/C++

**OJO:** Para que esto funcione, se deben identificar y resolver múltiples problemas de protocolo, de bloqueos, y otros.

**3. Entregables**

**Primera entrega: viernes 29 de abril de 2019**

Diseño de la solución. ¡Lo más gráfico posible! **OJO:** No queremos ver código. Pueden escribirlo (dibujarlo) a mano con las anotaciones necesarias. Traer a la clase para entregar. No se olviden de identificar el grupo.

**Segunda entrega: martes 2 de abril de 2019**

Únicamente lo correspondiente a procesos locales, con el emisor reconstruyendo las fotos en otro directorio, en lugar de enviarlas remoto.

**Entregables específicos:**

- Diseño (mejorado) de la solución
- Especificación del protocolo de comunicación
- Diagrama con tiempos y posibles estados de los actores
- Descripción de pruebas realizadas (*testing*).
- Dificultades enfrentadas durante el desarrollo de la solución y los mecanismos utilizados para enfrentarlos y resolverlos.
- Presentación (mínimo 6 páginas), con dos páginas de segmentos relevantes del código que muestren lo elegante de su solución.
- Código fuente

**Tercera entrega: jueves 4 de abril de 2019**

El enunciado completo.

**Entregables específicos:**

- Diseño (final) de la solución.
- Especificación (final) del protocolo de comunicación.
- Diagrama (final) con tiempos, **localidad**, y posibles estados de los actores.
- Comparación entre la solución por hilos y por procesos independientes de los subcontratistas.
- Descripción de pruebas realizadas (*testing*).
- Dificultades enfrentadas durante el desarrollo de la solución y los mecanismos utilizados para enfrentarlos y resolverlos.
- Presentación (mínimo 6 páginas), con dos páginas de segmentos selectos de código
- Código fuente