

## Sistemas Operativos (SIU4085)

### Proyecto

30%

## Simulación de un Sistema de Reservas

### Objetivos del Proyecto

- a. Resolver un problema concurrente utilizando procesos y los hilos de la librería POSIX.
- b. Comunicar los procesos utilizando pipes.
- c. Uso de señales para realizar operaciones periódicas y comunicación asíncrona entre procesos.
- d. Familiarizarse con llamadas al sistema para la gestión de procesos e hilos y el manejo de archivos

### Descripción General



Debido a la situación de pandemia que se vive desde el año pasado, el gobierno ha decretado medidas para la protección de los ciudadanos, y al mismo tiempo está fomentado actividades para activar sectores importantes como el comercio y el turismo. Una de esas medidas es establecer un aforo máximo para los sitios que usualmente visitan los turistas como las playas o parques nacionales. Esto ha obligado a los gobiernos y entidades locales, a desarrollar aplicaciones para que las personas reserven por anticipado su entrada a estos lugares.

En este proyecto vamos a desarrollar una aplicación para la reserva de espacios en una playa pública. Adicionalmente simularemos su ejecución con las familias que entrarán a usar el servicio en un determinado día.

La aplicación tendrá una arquitectura del tipo cliente/servidor. El servidor, llamado **Controlador de Reserva**, es un proceso que atenderá las solicitudes de reserva para estar en la playa. Estas solicitudes son realizadas por varios **Agentes de Reserva**, que son los procesos cliente. Los procesos se comunicarán a través de pipes nominales. A continuación, se explican de forma

general las funciones de estos componentes para posteriormente exponer detalles de su implementación.

## Controlador de Reserva

Este proceso tiene las siguientes funciones:

- Recibe las solicitudes de los Agentes de Reserva y las autoriza o rechaza dependiendo de la cantidad de gente que ya ha reservado en las horas solicitadas. En algunos casos, si es posible, coloca a las personas en un espacio distinto al solicitado.
- Va simulando el tiempo transcurrido. Y cada vez que transcurre una “hora” realiza las siguientes acciones: saca personas de la playa porque se les acaba su tiempo y autorizar la entrada de las personas que reservaron para la siguiente hora.
- Al finalizar, emite un reporte acerca de la ocupación de la playa.

## Agentes de Reserva

Se podría decir que son agencias de viaje donde los usuarios van a gestionar las reservas. Los agentes son procesos que solicitan al Controlador reservas para grupos familiares en un determinado día.

# Implementación de los procesos

## Controlador de Reserva

El proceso se invoca desde el Shell de la siguiente forma:

\$ ./controlador -i horaInicio -f horafinal -s segundoshora -t totalpersonas -p pipeprecibe

Donde:

- **horainicio**: hora inicial para comenzar la simulación. Las horas estarán en formato de 24 horas (1 -24). Se supone que la playa está abierta desde las 7 hasta las 19 horas, por lo que la hora de inicio de la simulación debe estar en esos rangos y debe ser menor a las 19 horas. Las horas serán números enteros.
- **horafinal**: hora final de la simulación. Las horas estarán en formato 1-24. Se supone que la playa está abierta desde las 7 hasta las 19 horas, por lo que una hora que se encuentre

fuera de estos rangos no debe ser válida. Las horas serán números enteros y la hora final de simulación debe ser mayor a la hora inicial.

- **segundoshora:** para disminuir el tiempo de simulación, este parámetro indicará el número en segundos de tiempo real que deben transcurrir para simular que ha pasado una hora en la simulación.
- **totalpersonas:** Es la cantidad máxima de personas que pueden estar en la playa en cada hora (aforo permitido)
- **pipeprecibe:** Por este pipe, los agentes envían información al Controlador de Reserva.

A continuación, se explican las diferentes funciones de este proceso:

**Al momento de inicialización:** el proceso inicializa la hora de su reloj según el parámetro recibido (**horainicio**), inicializa sus estructuras de datos, realiza las instrucciones correspondientes para que el reloj le avise cuando ha transcurrido una hora (dentro de los próximos **segundoshora**), crea y abre el pipe **pipeprecibe** para las comunicaciones con los agentes y se dispone a esperar, tanto la conexión de los agentes, como la señal del reloj.

**Mientras simula un día de playa:** pueden ocurrir los siguientes eventos:

- **Conexión de un agente:** Cada vez que se crea un agente, éste le envía un mensaje inicial al controlador donde le escribe su nombre (ver parámetros de entrada del agente). El controlador le devuelve la hora actual del sistema con base en la simulación que está realizando. Esta comunicación inicial puede usarse también para que el agente le envíe al controlador el nombre del pipe por donde se enviarán datos en el sentido controlador->agente y cualquier otra información que se considere pertinente.
- **Cuando se reciben peticiones de los agentes:** las peticiones de los agentes estarán relacionadas únicamente con la reserva para un grupo familiar de X personas que desean estar en la playa dos horas a partir de la hora W. El controlador o un hilo del controlador, recibe la petición, revisa sus estructuras y puede devolver la siguiente respuesta al Agente:
  - **Reserva ok.** La solicitud es aprobada, las personas pueden entrar en la playa por **2 horas** a partir de la hora solicitada. La hora solicitada se encuentra dentro del periodo de simulación.
  - **Reserva garantizada para otras horas.** Para las horas solicitadas por el agente ya no hay cupo en la playa, pero hay cupo en otras horas durante el periodo de simulación. En este caso, el controlador responde con la nueva hora donde ha colocado a la familia.
  - **Reserva negada por tarde.** En este caso el agente está solicitando una reserva para una hora que ya pasó. Al igual que en la situación anterior, si

hay algún cupo de dos horas posteriores (dentro del rango de la simulación), se procede a realizar la reserva en horas diferentes a las solicitadas.

- **Reserva negada, debe volver otro día.** El controlador negará la solicitud de reserva y no propondrá nuevas opciones cuando: a). La hora solicitada sea mayor a **horafinal**. b) El agente ha solicitado una determinada hora y el controlador no encuentra disponible ningún bloque de dos horas dentro del periodo de simulación donde se pueda ubicar a la familia. c) El número de personas incluidas en la reserva es mayor al aforo permitido.

El controlador debe imprimir las peticiones que va recibiendo en la pantalla (agente que las envía, hora solicitada y cantidad de personas). En la consola del agente, también se deben poder observar las peticiones y sus respectivas respuestas.

- **Transcurre una hora:** Cada vez que ha transcurrido una “hora de simulación” el controlador imprime la hora actual en formato 1-24, también indica cuántas

personas y qué familias salen y entran de la playa. Finalmente, avanza el reloj y coloca la alarma para la próxima hora.

**Al finalizar la simulación:** al finalizar la simulación según los parámetros establecidos, el controlador imprimirá un reporte que contiene:

- a) Horas pico: horas del día en las cuales la playa tuvo el mayor número de personas.
- b) Hora u horas del día en las cuales la playa tuvo el menor número de personas.
- c) Número de solicitudes negadas
- d) Número de solicitudes aceptadas en su hora
- e) Número de solicitudes reprogramadas

El controlador debe terminar a los Agentes con una señal o debe enviarles un mensaje para que terminen.

## Agentes

Este tipo de proceso se invoca de la siguiente forma:

\$ agente -s *nombre* -a *archivosolicitudes* -p *pipecrecibe*

Donde:

- *nombre*: es nombre del Agente. No es más que una cadena de caracteres suministrada por el usuario para identificar a un agente particular.
- *archivosolicitudes*: es el archivo de entrada de donde cada agente tomará las solicitudes de reserva. Se trata de un archivo de texto que contiene en cada línea una solicitud de reserva con el siguiente formato:

```
nombrefamilia,hora,#personas
```

El primer componente es una cadena de caracteres para identificar la familia que hace la reserva, el segundo es la hora (7-19) representada por un numero entero, y el último componente de la línea es el número de personas solicitando la reserva. Cada reserva es por un total de dos horas. Los campos están separados por comas. A continuación, un ejemplo del contenido del archivo:

```
Zuluaga,8,10  
Perez,8,4  
Rojas,10,10
```

El agente lee una línea, envía la solicitud al controlador, espera la respuesta y la imprime. Luego, envía al Controlador la siguiente solicitud de reserva. Dado que el agente, al

registrarse con el controlador, recibe la hora actual de simulación, no debe enviar solicitudes con una hora anterior a la hora actual.

- *pipecrecibe*: pipe nominal para la comunicación inicial de los agentes con el controlador. Es el mismo pipe que recibe el Controlador como parámetro en su creación.

## Detalle de las funciones del Agente

**Invocación y registro con el controlador:** los procesos agentes se pueden iniciar en cualquier momento, una vez que ha sido inicializado el Controlador de Reservas. Al iniciarse un agente, abre el pipe de comunicación con el controlador (*pipecrecibe*) y le envía su nombre para que éste lo registre. Es posible que le envíe información adicional según el diseño que realice cada grupo, como por ejemplo, el nombre del pipe para recibir las respuestas del controlador.

**Lectura del archivo y solicitudes de reserva:** Una vez registrado con el controlador, el agente lee el archivo de entrada y ejecuta el siguiente algoritmo:

1. Lee línea del archivo.
2. Valida que la hora de reserva no sea inferior a la hora actual de simulación.
3. Envía la solicitud y espera respuesta.
4. Imprimir la respuesta: positiva, reprogramación o denegada.
5. Esperar *tesolicitud* segundos y volver al paso 1.

### Fin de Archivo:

Cuando el archivo termina, el agente termina dando el siguiente mensaje:

Agente <nombre> termina.

## Primera Entrega

Valor de la entrega: 30% (6/20ptos) 3 ptos el informe y 3pts las funciones solicitadas.

La primera entrega se realizará el día 26 de abril y la sustentación el día 27 de abril.

Para la primera entrega debe tener lista las siguientes funcionalidades:

- Activación del Controlador de Reserva
- Activación y registro de al menos un agente.
- El agente le envía las solicitudes de reserva que consigue en su archivo al controlador y el controlador las registra en su estructura de datos. El controlador puede aceptar, negar

y reprogramar solicitudes. Para la primera entrega NO se pide la simulación de las horas del día.

- Después de que termine el agente, el controlador imprime cuántas personas quedaron reservadas para cada hora del día.

Adicionalmente, los estudiantes deben entregar:

- Un informe donde se explique (apoyado con gráficos): a) Estructuras de datos utilizadas en el proceso controlador para realizar las reservas. b) Diseño interno del controlador. c) Mecanismos de comunicación: número de pipes entre el controlador y los agentes. d) Formato de los mensajes que serán enviados entre los Agentes y el Controlador (en ambos sentidos) .
- Códigos fuente y un archivo makefile para generar el ejecutable. Todo debe venir en un archivo tar.gz.

## Entrega Final

La entrega final será el día 25 de mayo y la sustentación el día 26. Consiste de los códigos fuente del proyecto y un archivo makefile colocados en un archivo en formato tar.gz . **Sin excepción, los proyectos que no se entreguen el día 25 no serán calificados.** El día de la sustentación los profesores indicarán los archivos de entrada y los casos de prueba.

### Observaciones Adicionales

- El proyecto lo deben realizar en grupos de como máximo 3 estudiantes.
- Tanto para la primera, como para la segunda entrega, todos los integrantes del equipo deben estar presentes a la hora de la sustentación. Cualquier problema con el horario de la sustentación debe avisarse en los días previos a la misma.
- Cualquier duda sobre el enunciado del proyecto debe consultarla con los profesores en forma oportuna. La comprensión del problema y su correcta implementación, según lo indica el enunciado, es parte de lo que se está evaluando.
- Los grupos pueden discutir e intercambiar ideas de forma verbal, **pero bajo ningún concepto pueden compartir código o resultados del informe. Si se detecta copia en los productos entregados, los integrantes de los grupos serán citados a la Dirección de Carrera y el caso será elevado al Decano de la Facultad.**

Suerte

Profs. Mariela Curiel y Ricardo González.