

## 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. Utilizar señales para realizar operaciones periódicas.
- 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, pero al mismo tiempo esta 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 posibles 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 las funciones

de estos componentes para posteriormente exponer detalles sobre sus parámetros y protocolos de comunicación.

## 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, presente en el lugar, en la hora solicitada. En algunos casos, al rechazar una reserva, puede proponer una nueva hora.
- Va simulando el tiempo transcurrido. Y cada vez que transcurre una hora realiza las siguientes acciones: sacar personas de la playa (porque se les acaba su tiempo) y autorizar la entrada de las personas que reservaron para una determinada hora.
- Al finalizar, emite un reporte acerca de las horas de mayor y menor 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 envían al Controlador de Reserva información sobre reservas para un determinado día. En la aplicación pueden existir varios agentes de este tipo.

## Implementación de los dos tipos de procesos

### Controlador de Reserva

#### Parámetros de Entrada

El proceso se invoca 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 una hora que se encuentre fuera de estos rangos no debe ser válida para reserva. 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 debe ser mayor a la hora inicial.

- **segundoshora:** para disminuir el tiempo de simulación, y que ésta no dure un día completo, 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 dada la situación actual.
- **pipeprecibe:** pipe nominal para la comunicación inicial de los agentes al controlador. 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 las estructuras de datos para registrar el número de personas que están en la playa en cada hora, realiza las instrucciones correspondientes para que el reloj le avise cuando ha transcurrido una hora (dentro de los próximos *segundoshora*) y se dispone a esperar la conexión de los agentes.

**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). El controlador creará un hilo para atender a cada nuevo agente. Esta comunicación inicial puede usarse para que el Agente le envíe al controlador el nombre del pipe por el cual el Controlador le enviará información.
- **Cuando se reciben peticiones de los agentes:** las peticiones de los agentes estarán relacionadas únicamente con la reserva para un grupo 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.
  - **Reserva negada, porque se ha llenado el aforo,** se propone nueva hora: para la hora solicitada por el agente ya no hay cupo en la playa, pero hay cupo a otras horas durante el día. En este caso el Controlador responde proponiendo una nueva hora.
  - **Reserva negada por tarde:** en este caso el Agente está solicitando una reserva para una hora que ya pasó. En este caso, si hay algún cupo de dos horas en una hora posterior, se procede igual al caso anterior, es decir, se propone una nueva hora.
  - **Reserva negada, debe volver otro día.** El controlador negará la solicitud de reserva y no propondrá nuevas opciones cuando: a). La hora sea mayor a las 17 horas (la última reserva de dos horas se puede hacer a las 17). b) El

Agente ha solicitado una determinada hora y el controlador no encuentra cupo a ninguna hora en el día. c) El número de personas incluidas en la reserva es mayor al aforo permitido.

El Controlador debe imprimir todas las peticiones recibidas en su pantalla (Agente que las envía, hora solicitada y cantidad de personas), tanto las peticiones, como sus respectivas respuestas se debe ver en la consola del agente correspondiente)

- **Transcurre una hora:** Cada vez que ha transcurrido una “hora de simulación” el Controlador imprime la hora actual en formato 1-24. También indica cuantas y cuales personas salen de la playa y cuantas y cuales entran (realmente se muestra es el nombre de la familia con la cuál esta la reserva).

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

- a) Hora (u horas pico): horas en las cuales la playa tuvo el mayor número de personas.
- b) Hora u horas en las cuales la playa tuvo el menor número de personas.
- c) Número de solicitudes negadas
- d) Número de solicitudes aceptadas

## Agentes

### Parámetros de Entrada

Este tipo de proceso se invoca de la siguiente forma:

\$ agente -s *nombre* -a archivosolicitudes -t tesolicitud -p pipeprecibe

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. Puede existir o

no un espacio entre la coma y el valor numérico, su programa debe funcionar en ambos casos. A continuación, un ejemplo:

```
Zuluaga, 7, 10
Perez, 7, 4
Rojas, 8, 10
, ,
```

El Agente lee una línea, envía la solicitud al controlador, espera respuesta (y actúa conforme a la respuesta) y espera un tiempo *tesolicitud* para enviar 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. El archivo debe terminar con una línea que inicia con dos comas consecutiva

- *tesolicitud*: Este es un número entero que representa el tiempo en segundos que espera el agente para leer la próxima solicitud del archivo y enviarla al controlador, el valor puede ser 0, en cuyo caso los datos se enviarían inmediatamente
- *pipeprecibe*: pipe nominal para la comunicación inicial de los Agentes con el Controlador.

### Detalle de las funciones del Agente

**Invocación y registro con el Controlador:** los procesos Agente se pueden iniciar en cualquier momento, una vez inicializado el controlador de reservas. Al iniciarse un Agente, abre el pipe de comunicación con el Controlador (*pipeprecibe*) le envía su nombre al Controlador para que este lo registre (y posiblemente otra información) y comienza a leer y a enviar las solicitudes de reserva que se encuentran en un archivo.

**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. Si la respuesta es ok, ir al siguiente paso. Si la respuesta es reprogramación, aceptar la sugerencia del controlador e ir al siguiente paso. Si la respuesta es denegada ir al siguiente paso.
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: 14% (5% el informe y 9% las funciones solicitadas)

La primera entrega se realizará el día Jueves 27 de abril y la sustentación será el 29 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 o 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.
- 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, número de hilos y funciones. 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)

## Observaciones Adicionales

- El proyecto lo deben realizar en grupos de 3 estudiantes.
- La entrega final la deben realizar el día 25 de mayo antes de las 12 de la noche por UVirtual. **Sin excepción, los proyectos que no estén a esa hora en el UVirtual no serán calificados.** La sustentación será el jueves 29 de mayo.
- Las sustentaciones de la primera y segunda entrega se realizarán en la fecha que la profesora indique. Cada entrega debe tener los códigos fuentes y el makefile. En la primera entrega deberán adicionalmente entregar un informe con el diseño. Todos los integrantes del equipo deben estar 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.