

SEGUNDO PROYECTO
SISTEMAS OPERATIVOS



ESTUDIANTES:

Juan David Castillo

Juan Sebastian Clavijo

Juliana Lugo

Juan Camilo Martínez

Santiago Mesa

Juan Daniel Vargas

DOCENTE:

John Jairo Corredor

Facultad de Ingeniería

FECHA DE ENTREGA:

20 de noviembre de 2023

Introducción: Este proyecto desarrolla un sistema de reservas para el parque Berlín, orientado a controlar el aforo y mejorar la experiencia de los visitantes.

OBJETIVO: Implementar un sistema que permita hacer reservas por tiempo en un parque privado utilizando una arquitectura cliente/servidor en una sola máquina. La idea de este proyecto es emplear todo lo aprendido a lo largo del semestre haciendo un énfasis en la utilización y manejo de *pipes*.

Arquitectura Cliente/Servidor: Se utilizará una arquitectura con los siguientes componentes:

- **Controlador de Reserva (Servidor):** Es el que se encarga de gestionar las solicitudes de reserva y simula el paso del tiempo en el parque.
- **Agentes de Reserva (Clientes):** Son aquellos que se encargan de solicitar los espacios para grupos familiares.

Funcionalidades específicas

Controlador de reserva:

- Recibe y gestiona las solicitudes de reserva.
- Simula el tiempo transcurrido en el parque.
- Emite un reporte al final del día con detalles de la ocupación.

Agentes de Reserva:

- Realizan solicitudes de reserva.
- Se comunican con el controlador a través de un pipe.

Implementación

- **Controlador de Reserva:** Se inicia con parámetros específicos para la simulación (hora de inicio, hora final, segundos por hora, total de personas y nombre del pipe).
- **Agentes de Reserva:** Se inician con un nombre, un archivo de solicitudes y un pipe para la comunicación inicial.

Marco Teórico

- **Pipes:** Se pueden definir como un mecanismo de comunicación y/o sincronización entre procesos en sistemas operativos. En C, se utilizan para establecer una comunicación entre procesos al crear un flujo de datos unidireccional entre ellos.
- **Modularización:** Se refiere al enfoque de dividir un programa grande y complejo en pequeños módulos más pequeños y manejables. Estos módulos son un tipo de unidades independientes y cohesivas que realizan las tareas específicas dentro del programa. En este caso se manejan mediante Funciones, archivos de encabezado, ('.h - .p'). Estos contienen declaraciones de funciones, estructuradas y variables utilizadas en el archivo del código fuente ('.c'). Esto permite la reutilización de funciones y facilita la organización del código ya que separa la interfaz pública de la implementación privada,

Funcionamiento del sistema:

- **Agente:** Este se encarga de ser el “cliente” en el sistema. Cuenta con una estructura de datos que contiene el nombre, la hora, la cantidad de personas y la respuesta que recibe para ingresar al parque. El agente se encarga de recibir el siguiente comando en la línea de comandos: *./agente -s nombre -a archivosolicitudes -p pipeprecibe*. Si los datos ingresados cumplen con los requisitos, entonces el agente asigna cada valor a cada dato dentro de la estructura definida con anterioridad. Una vez definido esto, procede a leer los datos que se encuentren dentro del archivo especificado que deben venir en el siguiente formato:

Zuluaga,8,10

Perez,8,4

Rojas,10,10

Por cada línea que se lee en el archivo, se procede a enviar esta por medio del pipe especificado con la estructura esperada. Posterior a esto el agente espera la respuesta del servidor para luego procesarla. Al finalizar se cierra el pipe utilizado y el archivo desde donde vienen las solicitudes.

- **Controlador de Reserva:** Por otro lado, se tiene al controlador de reserva que simula ser el “servidor” del sistema, el cual cabe resaltar que trabaja en su mayoría con hilos, para la recepción y emisión de mensajes. Cuenta con 3 estructuras de datos para almacenar los datos de espera, los datos del pipe y la solicitud recibida. Para iniciar el servidor, se debe mandar la siguiente línea de comando: *./controlador -i horaInicio -f horafinal -s segundoshora -t totalpersonas -p pipeprecibe -c horaCierre*.

Una vez comienza a ejecutarse, el servidor se encarga de simular el tiempo que pasa en el parque con base al parámetro enviado como segHora, es decir, si recibe por ejemplo un valor de 20, cada 20 segundos el parque aumenta 1 hora. Este servidor gestiona las solicitudes que le llegan a través del pipe, en esta parte, se encarga de

validar ya sea la cantidad de personas permitidas en el parque o la hora a la que se quiere ingresar, dependiendo de las condiciones genera una respuesta, ya sea aceptada la reserva, denegada o reprogramada. Al finalizar el día, el servidor muestra una serie de estadísticas como la cantidad de personas promedio, la cantidad de personas denegadas o reprogramadas o la hora más concurrida para así dar por finalizada la ejecución.

PLAN DE PRUEBAS:

Plan de pruebas:			
Descripción de caso	Valores de entrada	Resultado esperado	Resultado obtenido
1. Menos argumentos de los requeridos en controlador	Por parte del controlador se requieren 11 argumentos	Imprime línea de código donde dice que faltan argumentos	Ver figura 1.
2. Menos argumentos de los requeridos en agente	Por parte del controlador se requieren 9 argumentos	Imprime línea de código donde dice que faltan argumentos	Ver figura 2.
3. Funcionamiento con parámetros en desorden	Sin importar el orden en el que se escriban los comandos, tanto en controlador como el agente, toma los valores correspondientes y los asigna de la forma correspondiente.	El programa asigna cada valor a cada variable correspondiente teniendo en cuenta la letra y el valor siguiente a esta	Ver figura 3.
4. Hora inicial mayor a hora final	El sistema no permite que el parque se abra a una hora superior de la hora a la que cierra	El programa presenta por pantalla que la hora final debe ser mayor a la hora inicial	Ver figura 4.
5. Ejecución con un agente	Se realiza la ejecución utilizando un	El programa se ejecuta de manera exitosa y procesa las solicitudes de un	Ver Figura 5.

	solo agente con 1 único archivo	solo agente, enseñando cuales peticiones fueron aceptadas, rechazadas o reprogramados	
6. Ejecución con 2 agentes	Se realiza la ejecución utilizando 2 terminales (simulando 2 agentes) utilizando 2 documentos distintos	El programa se ejecuta de manera exitosa y procesa las solicitudes de 2 agentes, enseñando cuales peticiones fueron aceptadas, rechazadas o reprogramados	Ver figura 6.

Evidencia de los resultados obtenidos

Figura 1.

```

jclavijo@SO: ~/Desktop/Operativos
jclavijo@SO:~/Desktop/Operativos2330/proyecto2$ ./controlador
Uso: ./controlador -i horaInicio -f horafinal -s segundoshora -t totalpersonas -p pipecrecibe
jclavijo@SO:~/Desktop/Operativos2330/proyecto2$ ./controlador -i 7 -s 13 -t 40 -p pipe
Uso: ./controlador -i horaInicio -f horafinal -s segundoshora -t totalpersonas -p pipecrecibe
jclavijo@SO:~/Desktop/Operativos2330/proyecto2$

```

Figura 2.

```

jclavijo@SO:~/Desktop/Operativos2330/proyecto2$ ./agente
Uso: ./agente -s nombre -a archivosolicitudes -p pipecrecibe -c horacierre
jclavijo@SO:~/Desktop/Operativos2330/proyecto2$ ./agente -s perry -a prueba.txt
Uso: ./agente -s nombre -a archivosolicitudes -p pipecrecibe -c horacierre
jclavijo@SO:~/Desktop/Operativos2330/proyecto2$

```

Figura 3.

```
jclavijo@SO: ~/Desktop/Operativos2330/proyecto2
total 80K
drwxrwxr-x 2 jclavijo jclavijo 4,0K nov 20 22:25 .
drwxrwxr-x 15 jclavijo jclavijo 4,0K nov 7 15:48 ..
-rwxrwxr-x 1 jclavijo jclavijo 17K nov 20 22:25 agente
-rw-rw-r-- 1 jclavijo jclavijo 3,7K nov 20 21:55 agente.c
-rw-rw-r-- 1 jclavijo jclavijo 34 nov 7 16:03 archivo.txt
-rw-rw-r-- 1 jclavijo jclavijo 335 nov 20 19:25 Comandos.txt
-rwxrwxr-x 1 jclavijo jclavijo 17K nov 20 22:25 controlador
-rw-rw-r-- 1 jclavijo jclavijo 6,8K nov 20 22:08 controlador.c
prw-rw-r-- 1 jclavijo jclavijo 0 nov 20 21:58 nuevo_pipe
-rw-rw-r-- 1 jclavijo jclavijo 130 nov 20 19:25 nuevo_pipe.lnk
-rw-rw-r-- 1 jclavijo jclavijo 398 nov 20 21:38 prueba2.txt
-rw-rw-r-- 1 jclavijo jclavijo 369 nov 20 21:37 prueba.txt
prw-rw-r-- 1 jclavijo jclavijo 0 nov 7 16:04 reservation_pipe
jclavijo@SO:~/Desktop/Operativos2330/proyecto2$ ./controlador -i 7 -s 13 -t 40 -p nuevo_pipe -f 19

Hora de inicio: 7
Hora final: 19
Segundos por hora: 13
Total de personas: 40
Nombre del pipe: nuevo_pipe

Hora de inicio: 7
Hora final: 19
```

Figura 4.

```
jclavijo@SO: ~/Desktop/Operativos2330/proyecto2$ ./controlador -i 14 -s 13 -t 40 -p nuevo_pipe -f 13
La hora final debe ser mayor que la hora de inicio.
jclavijo@SO:~/Desktop/Operativos2330/proyecto2$
```

Figura 5.

```
Solicitud denegada: Soto, Hora: 10, Personas: 6 (Hora inválida)
Hora Actual: 13
Solicitud autorizada: Hernandez, Hora: 14, Personas: 4
Solicitud autorizada: Castro, Hora: 16, Personas: 7
Solicitud autorizada: Rivera, Hora: 18, Personas: 5
Hora Actual: 14
Solicitud denegada: Alvarez, Hora: 11, Personas: 9 (Hora inválida)
Solicitud denegada: Gonzalez, Hora: 13, Personas: 3 (Hora inválida)
Hora Actual: 15
Solicitud autorizada: Romero, Hora: 15, Personas: 12
Solicitud autorizada: Mendoza, Hora: 17, Personas: 8
Solicitud denegada: Suarez, Hora: 9, Personas: 5 (Hora inválida)
Hora Actual: 16
Solicitud denegada: Navarro, Hora: 12, Personas: 7 (Hora inválida)
Solicitud denegada: Vargas, Hora: 14, Personas: 10 (Hora inválida)
Hora Actual: 17
Solicitud denegada: Jimenez, Hora: 16, Personas: 4 (Hora inválida)
Solicitud autorizada: Reyes, Hora: 18, Personas: 6
Solicitud denegada: Ortega, Hora: 10, Personas: 9 (Hora inválida)
Hora Actual: 18
Solicitud denegada: Acosta, Hora: 7, Personas: 8 (Hora inválida)
Hora Actual: 19
El hilo de solicitudes se ha cerrado

Reporte final:
Total de personas en el parque: 135
Hora 7: 0 personas
Hora 8: 0 personas
Hora 9: 7 personas
Hora 10: 0 personas
Hora 11: 0 personas
Hora 12: 0 personas
Hora 13: 0 personas
Hora 14: 0 personas
Hora 15: 12 personas
Hora 16: 0 personas
Hora 17: 0 personas
Hora 18: 0 personas
Hora 19: 0 personas

Personas promedio por hora: 1
Hora más concurrida: 15 con 12 personas
Solicitudes denegadas: 10
Solicitudes reprogramadas: 0
jclavijo@SO:~/Desktop/Operativos2330/proyecto2$

Nombre: Diaz, Hora: 11, Número de personas: 12
Respuesta: Denegada
Nombre: Ramirez, Hora: 13, Número de personas: 5
Respuesta: Denegada
Nombre: Torres, Hora: 15, Número de personas: 9
Respuesta: Denegada
Nombre: Gutierrez, Hora: 17, Número de personas: 6
Respuesta: Denegada
Error: La hora de reserva es anterior a la hora actual de simulación o después del cierre.
Nombre: Luna, Hora: 16, Número de personas: 8
Respuesta: Denegada
Nombre: Moreno, Hora: 7, Número de personas: 10
Respuesta: Denegada
Nombre: Soto, Hora: 10, Número de personas: 6
Respuesta: Denegada
Nombre: Hernandez, Hora: 14, Número de personas: 4
Respuesta: Denegada
Nombre: Castro, Hora: 16, Número de personas: 7
Respuesta: Denegada
Nombre: Rivera, Hora: 18, Número de personas: 5
Respuesta: Denegada
Nombre: Alvarez, Hora: 11, Número de personas: 9
Respuesta: Denegada
Nombre: Gonzalez, Hora: 13, Número de personas: 3
Respuesta: Denegada
Nombre: Romero, Hora: 15, Número de personas: 12
Respuesta: Denegada
Nombre: Mendoza, Hora: 17, Número de personas: 8
Respuesta: Denegada
Nombre: Suarez, Hora: 9, Número de personas: 5
Respuesta: Denegada
Nombre: Navarro, Hora: 12, Número de personas: 7
Respuesta: Denegada
Nombre: Vargas, Hora: 14, Número de personas: 10
Respuesta: Denegada
Nombre: Jimenez, Hora: 16, Número de personas: 4
Respuesta: Denegada
Nombre: Reyes, Hora: 18, Número de personas: 6
Respuesta: Denegada
Nombre: Ortega, Hora: 10, Número de personas: 9
Respuesta: Denegada
Nombre: Acosta, Hora: 7, Número de personas: 8
Respuesta: Denegada
Agente jameSB termina
jclavijo@SO:~/Desktop/Operativos2330/proyecto2$
```

Figura 6.

```
jclavijo@SO: ~/Desktop/Operativos2330/proyecto2
Solicitud denegada: Suarez, Hora: 10, Personas: 3 (Hora inválida)
Solicitud autorizada: Romero, Hora: 15, Personas: 12
Solicitud denegada: Torres, Hora: 8, Personas: 6 (Hora inválida)
Hora Actual: 16
Solicitud autorizada: Mendoza, Hora: 17, Personas: 8
Solicitud denegada: Moreno, Hora: 10, Personas: 7 (Hora inválida)
Solicitud denegada: Suarez, Hora: 9, Personas: 5 (Hora inválida)
Solicitud denegada: Rivera, Hora: 14, Personas: 5 (Hora inválida)
Solicitud denegada: Navarro, Hora: 12, Personas: 7 (Hora inválida)
Hora Actual: 17
Solicitud denegada: Ortega, Hora: 16, Personas: 10 (Hora inválida)
Solicitud denegada: Vargas, Hora: 14, Personas: 10 (Hora inválida)
Solicitud reprogramada: Castro, Hora: 10, Personas: 4 (Capacidad excedida)
Solicitud denegada: Jimenez, Hora: 16, Personas: 4 (Hora inválida)
Solicitud denegada: Jimenez, Hora: 9, Personas: 8 (Hora inválida)
Hora Actual: 18
Solicitud autorizada: Reyes, Hora: 10, Personas: 6
Solicitud denegada: Reyes, Hora: 12, Personas: 3 (Hora inválida)
Solicitud denegada: Ortega, Hora: 10, Personas: 9 (Hora inválida)
Solicitud denegada: Mendoza, Hora: 13, Personas: 6 (Hora inválida)
Hora Actual: 19
Solicitud denegada: Acosta, Hora: 7, Personas: 8 (Hora inválida)
El hilo de solicitudes se ha cerrado

Reporte final:
Total de personas en el parque: 181
Hora 7: 0 personas
Hora 8: 4 personas
Hora 9: 6 personas
Hora 10: 0 personas
Hora 11: 0 personas
Hora 12: 0 personas
Hora 13: 0 personas
Hora 14: 0 personas
Hora 15: 17 personas
Hora 16: 0 personas
Hora 17: 0 personas
Hora 18: 6 personas
Hora 19: 0 personas

Personas promedio por hora: 2
Hora más concurrida: 15 con 17 personas
Solicitudes denegadas: 25
Solicitudes reprogramadas: 2
jclavijo@SO: ~/Desktop/Operativos2330/proyecto2$

jclavijo@SO: ~/Desktop/Operativos2330/proyecto2
Respuesta: Denegada
Nombre: Romero, Hora: 15, Número de personas: 12
Respuesta: Denegada
Nombre: Mendoza, Hora: 17, Número de personas: 8
Respuesta: Denegada
Nombre: Suarez, Hora: 9, Número de personas: 5
Respuesta: Denegada
Nombre: Navarro, Hora: 12, Número de personas: 7
Respuesta: Denegada
Nombre: Vargas, Hora: 14, Número de personas: 10
Respuesta: Denegada
Nombre: Jimenez, Hora: 16, Número de personas: 4
Respuesta: Denegada
Nombre: Reyes, Hora: 18, Número de personas: 6
Respuesta: Denegada
Nombre: Ortega, Hora: 10, Número de personas: 9
Respuesta: Denegada
Nombre: Acosta, Hora: 7, Número de personas: 8
Respuesta: Denegada
Agente jamesb termina
jclavijo@SO: ~/Desktop/Operativos2330/proyecto2$ ./agente -s jamesb -a prueba.txt -p nuevo.p
ipe -c 19

jclavijo@SO: ~/Desktop/Operativos2330/proyecto2
Respuesta: Denegada
Nombre: Suarez, Hora: 10, Número de personas: 3
Respuesta: Denegada
Nombre: Torres, Hora: 8, Número de personas: 6
Respuesta: Denegada
Nombre: Moreno, Hora: 10, Número de personas: 7
Respuesta: Denegada
Nombre: Rivera, Hora: 14, Número de personas: 5
Respuesta: Denegada
Nombre: Ortega, Hora: 16, Número de personas: 10
Respuesta: Denegada
Nombre: Castro, Hora: 18, Número de personas: 4
Respuesta: Denegada
Nombre: Jimenez, Hora: 9, Número de personas: 8
Respuesta: Denegada
Nombre: Reyes, Hora: 12, Número de personas: 3
Respuesta: Denegada
Nombre: Mendoza, Hora: 13, Número de personas: 6
Respuesta: Denegada
Nombre: Vargas, Hora: 15, Número de personas: 9
jclavijo@SO: ~/Desktop/Operativos2330/proyecto2$
```

Referencias

- Paszniuk, R. (s. f.). *Pipes en C – Linux – programación*.
<https://www.programacion.com.py/escritorio/c/pipes-en-c-linux>
- *IBM documentation*. (s. f.).
<https://www.ibm.com/docs/es/was/8.5.5?topic=applications-modularization-challenge>
- Universidad Nacional San Luis. (s. f.). *Lenguaje C - Modularización Funciones* [Diapositivas; PDF].
<http://servicios.dirinfo.unsl.edu.ar/abm/assets/uploads/materiales/746a3-tr6-modularizacion-c-1c-2022.pdf>