

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS

PROYECTO SISTEMAS OPERATIVOS

MARIELA CURIEL

ENTREGA 1

JUAN PABLO ORTIZ RUBIO

CARLOS MIGUEL SÁNCHEZ LORETO

PAULA JULIANA ROJAS NARANJO

ABRIL 26, 2021

Tabla de contenido

[Introducción 3](#_Toc70287494)

[Lista de figuras 4](#_Toc70287495)

[Estructuras de datos 5](#_Toc70287496)

[Diseño interno del Controlador 7](#_Toc70287497)

[Mecanismos de comunicación 10](#_Toc70287498)

[Formato de mensajes 11](#_Toc70287499)

# Introducción

En este informe se detalla y se ilustra las distintas funcionalidades y mecanismos realizados para la elaboración de la primera entrega de la clase *Sistemas operativos*. Donde el principal tema y herramienta utilizada fue Pipes para la comunicación de dos procesos, en este caso el proceso Controlador y el proceso Agente.

Se explican las estructuras de datos utilizadas para la información de los procesos, se ilustra el diseño de los procesos, la explicación del mecanismo de comunicación utilizado y el formato de mensajes definidos para dicha comunicación entre pipes.

# 

# Lista de figuras

Fig 1. Estructura de datos Reserva 5

Fig 2. Estructura de datos Data y Agente 6

Fig 3. Obtener el agente y sus pipes 7

Fig 4. Leer datos del pipe 8

Fig 5. Verificar y asignar reserva OK 8

Fig 6. Verificar y asignar reserva garantizada para otra hora 8

Fig 7. Verificar y asignar reserva negada 9

Fig 8. Enviar respuesta y finalizar el proceso de reserva 9

Fig 9. Pipes de comunicación 10

Fig 10. Formato de mensajes respuesta 11

Fig 11. Struct reserva, mensajeRespuesta 12

Fig 12. Comunicación de proceso terminado 12

# Estructuras de datos

Para la definición de las estructuras utilizadas en el proyecto, se pensó en la versatilidad y facilidad que se tendría al manejarlas.

Por un lado, en la estructura de datos de Reserva se planteó un agrupamiento con *struct,*como se puede observar en la figura 1, ya que, dentro de ella, existen distintos datos, como lo son:

* El nombre de la familia, definido en un arreglo, de tipo char.
* La hora en la que la familia quiere ingresar a la playa, de tipo int.
* El número de personas en la familia que ingresarán a la playa, de tipo int.
* El
* El mensaje que se definirá dependiendo de la situación que se presente en la hora escogida especificada en diferentes números de respuesta, de tipo int.

Texto

Descripción generada automáticamente

Fig. 1. Estructura de datos Reserva

Por otro lado, se definieron otros dos tipos *struct,* data y Agente, como se puede observar en la figura 2, en donde el nombre del pipe se almacenará en segundoPipe de tipo char, unido respectivamente a su id, llamado pid de tipo int.

Para la información de Agente, se especificó la siguiente información necesaria:

* El nombre del agente, en un arreglo de tipo char.
* El nombre del pipeReceptor, en un arreglo de tipo char.
* El nombre del pipeEmisor, en un arreglo de tipo char.
* El id del pipe, de tipo int.

Texto

Descripción generada automáticamente

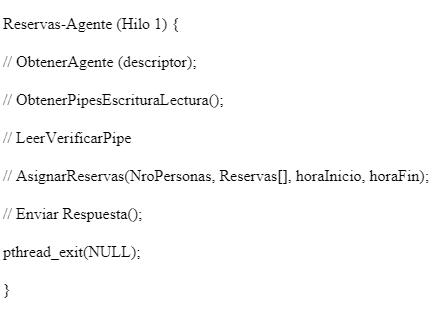
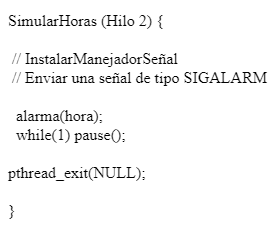
Fig. 2. Estructura de datos Data y Agente

# 

# Diseño interno del Controlador

El diseño interno del controlador será implementado mediante un sistema utilizando multihilos, que contará con un primer hilo para manejar las todas solicitudes enviadas por los agentes. A través de este hilo lo que se busca lograr es que se puedan administrar solicitudes de varios agentes al mismo tiempo, almacenar cada una de las reservas en una estructura de datos, enviar la respuesta correspondiente y lograr una mejor utilización del procesador.

También se incluye un segundo hilo para el manejo y simulación de las horas transcurridas durante el día y que utilizará adicionalmente señales de alarma.



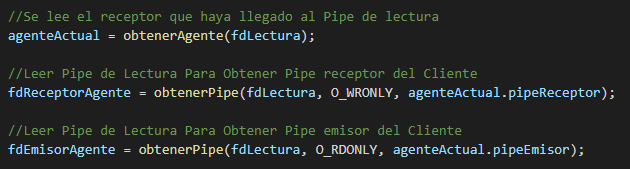


Fig. 3.Obtener el agente y sus pipes.

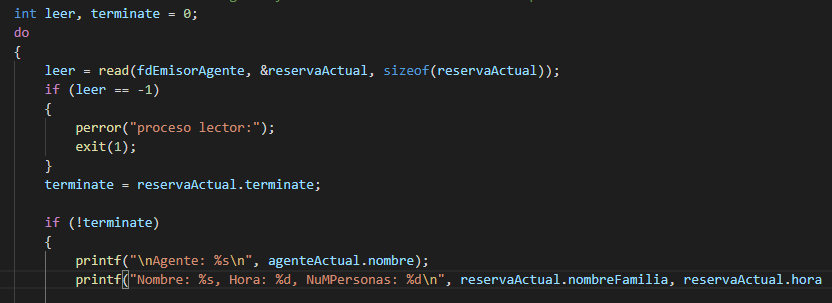


Fig. 4.Leer datos del pipe.

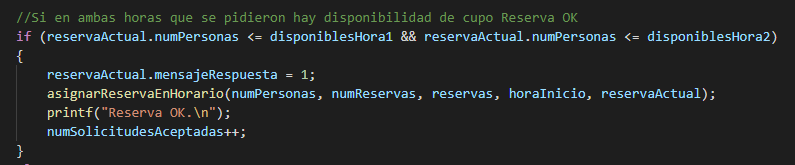


Fig. 5.Verificar y asignar **Reserva OK**.

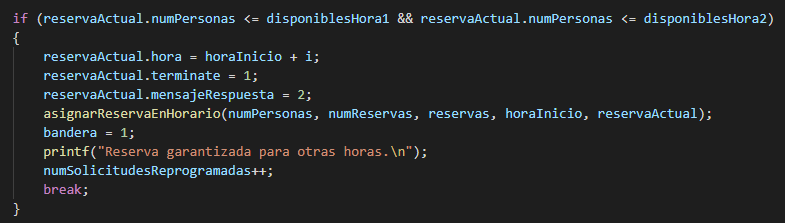


Fig. 6.Verificar y asignar **Reserva garantizada para otras horas**.

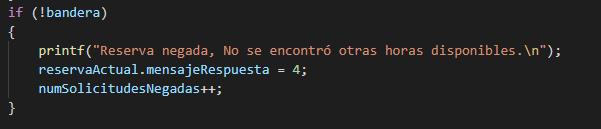


Fig. 7.Verificar y asignar **Reserva negada**.

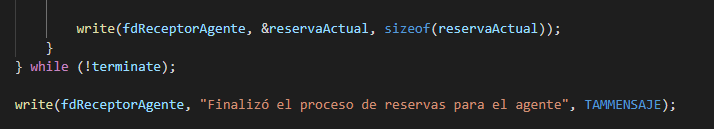


Fig. 8. Enviar respuesta y finalizar proceso de reserva.

# Mecanismos de comunicación

El controlador crea un pipe de lectura, llega un agente, en este caso Agente A, que se conecta al pipe del controlador donde escribe. Posteriormente le envía una estructura que tenga 2 pipes con su respectivo nombre y pid, en ese momento el pipe del controlador sigue abierto. Crea el Agente A, con dos pipes, el emisor y el receptor donde se envían por el pipe de lectura de controlador así conoce los pipes y se conecta.

El orden de creación de los pipes es el siguiente:

1. Controlador lectura.
2. Emisor y receptor del agente (Agente A).

Gráfico, Gráfico en cascada

Descripción generada automáticamente

Fig. 9. Pipes de comunicación

# 

# Formato de mensajes

Principalmente, para la comunicación de verificación de ciertos pasos entre pipes, se realizan envíos de mensajes entre ellos. Cuando una reserva es propuesta en el controlador, se debe verificar distintas opciones para asegurar que el aforo sea el adecuado en la hora propuesta y si esto no se cumple, encontrar la hora más cercana donde sí se pueda realizar la validación, pero si ninguna de las dos opciones anteriores se cumple, se debe enviar un mensaje de negación comunicando que la reserva no se puede realizar en el día propuesto.

Dichas opciones de mensaje de respuesta se detallan en la figura 4, donde se envía la reserva recién leída al controlador con la función *write* y se lee dicha respuesta dada por el controlador, con la función *read*.

Texto

Descripción generada automáticamente

Fig. 10. Formato de mensajes respuesta

Dichos mensajes de respuesta se encuentran en un *struct* de Reserva como se puede apreciar en la figura 5 donde la validación se guarda en la variable de tipo int mensajeRespuesta.

Texto

Descripción generada automáticamente

Fig. 11. Struct reserva, mensajeRespuesta

Para que el controlador se entere cuando el agente haya terminado el proceso de reserva, este envía una verificación que se guarda en la variable de tipo int terminate, que se evidencia en la figura 5.

Luego de la validación por parte del controlador respecto a las distintas variables que pueden afectar a la disponibilidad de reserva por parte del agente, este termina dicho ciclo de verificación hasta que la variable de Reserva, terminate, sea distinta a verdadero, o sea que haya terminado dicho ciclo.

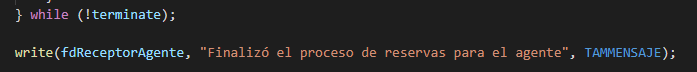


Fig. 12. Comunicación de proceso terminado

Cuando el agente termina la verificación de la reserva, el *while* acaba ya que cambia la variable terminate y justo después, el pipe controlador escribe con la función *write* en el pipe receptor del agente, que el proceso de reservas ha finalizado.