

## **ChismeGPT. Hailie Jiménez.**

### Documentación.

Core: Programa principal, contiene el servidor que recibe, agenda, y procesa a los clientes. La estructura de datos principal es la lista simplemente enlazada, y existen 3: postClientsQueue, waitingClientsQueue, y prepaidClientQueue. También, hay 2 arrays, uno de tipo User que almacena los usuarios que están en ejecución y otro de sem\_t para cada índice del arreglo de usuarios en ejecución, esto para poder ubicar usuarios o quitarlos sin condiciones de carrera.

Uso: ./core #numeroDeMensajesRecurrentes.

### Funciones principales:

1. Main: Mantiene al servidor activo, recibe los clientes y los ingresa a la cola dependiendo de su prioridad (20: prepago, 60: postpago). Crea los hilos (hasta #numeroDeMensajesRecurrentes) y llama a la función processRequirement que espera a recibir un cliente.
2. createUser: Crea al usuario apenas encuentra una nueva conexión al socket del servidor. El usuario envía el número de mensajes que va a enviar, el mensaje, el tipo de cliente que es.
3. Prioritize: Es un hilo aparte. Toma a los clientes de las respectivas colas, priorizando los clientes postpago, luego los de la cola de espera (clientes prepago que fueron movidos de su ejecución) y finalmente los clientes prepago.
4. Schedule: Es un hilo aparte. Toma el cliente listo para ubicar en el arreglo de clientes en ejecución, si es postpago tiene prioridad y puede reemplazar al primer prepago que encuentre en el arreglo. Los clientes prepago son reemplazados y ubicados en la cola waitingClientsQueue.
5. ProcessRequirement: Es la función llamada por los hilos creados en main, cada hilo en processRequirement recibe su índice en el array en ejecución, así cada hilo se enfoca en un índice en el array.

Multi\_Client: Genera n cantidad de clientes con ayuda del programa ./client que mande m número de mensajes.

Uso: ./multi\_client <num\_clientes> <num\_mensajes>

### Funciones principales:

1. Main: Genera n cantidad de clientes pasados como parámetro en el programa.
2. runClient: Crea un proceso y usa execvp para correr el programa ./client.

Client: Es el programa cliente individual, este envía el mensaje al servidor.

Uso: ./client -t <tipo> -n <numeroMensajes>

### Funciones principales.

1. Main: Genera dos hilos, uno que recibe mensajes del servidor y otro que los envía.
2. receiveMessages: Hilo dedicado a recibir e imprimir los mensajes del usuario.
3. SendMessages: Envía los mensajes al servidor. Envía el mensaje con el siguiente formato: numeroMsj|header message. Donde: numeroMsj es el número de mensajes que va a enviar el cliente, por default, es 2 (esta variable sirve como contador para simular el procesamiento en el servidor), header, para distinguir si es un cliente tipo PREPAGO o POSTPAGO, y message que es un mensaje definido como "Automated message" para simplificar.