

DISEÑO DE LA PRÁCTICA FINAL

Declaraciones GLOBALES:

- Semáforos y variables condición
 - **Fichero**, **colaClientes**, **ascensor**, **máquinas**
 - Condiciones para el ascensor y los clientes que deben subir
- Contador de clientes
- Número de clientes en ascensor
- Ascensor funcionando
- Lista de 20 clientes:
 - Id
 - Atendido
 - Tipo
 - Ascensor
- Recepcionistas (lista o no)
- MáquinasCheckIn (Como una lista)
- Fichero de log (FILE * logFile);

main {

1. signal o sigaction SIGUSR1, cliente normal.
2. signal o sigaction SIGUSR2, cliente vip.
3. signal o sigaction SIGINT, terminar.
4. Inicializar recursos (¡Ojo!, Inicializar!=Declarar).
 - a. Semáforos.
 - b. Contador de clientes.
 - c. Lista de clientes id 0, atendido 0, tipo 0, serología 0.
 - d. Lista de recepcionistas (si se incluye).
 - e. MáquinasDeCheckin (ponerlas todas como libres)
 - f. Fichero de Log
 - g. Variables relativas al ascensor.
 - h. Variables condición
5. Crear 3 hilos recepcionistas.
6. Esperar por señales de forma infinita.

}

nuevoCliente{

1. **Comprobar si hay espacio en la lista cola de clientes.**
 - a. **Si lo hay**
 - i. **Se añade el cliente.**
 - ii. **Contador de clientes se incrementa.**
 - iii. **nuevoCliente.id = ContadorClientes.**
 - iv. **nuevoCliente.atendido=0**
 - v. **tipo=Depende de la señal recibida y de si va o no a máquinas**
 - vi. **nuevoCliente.Ascensor=0.**
 - vii. **Creamos hilo para el cliente.**
 - b. **Si no hay espacio**
 - i. **Se ignora la llamada.**

}

AccionesCliente{

1. **Guardar en el log la hora de entrada.**
2. **Guardar en el log el tipo de cliente.**
3. **Si va a máquina de checkin comprueba si hay máquinas libres**
 - a. **Si las hay ocupa una máquina**
 - i. **Duerme 6 segundos (simulando la atención)**

- ii. Comprueba si va a ascensores
 - iii. Si va, ir a 6a.
 - iv. Si se fuera y **se escribe en el log**, se daría fin al hilo Cliente y **se liberaría el espacio en la cola**.
- b. Si no las hay
 - i. Duerme 3 segundos
 - ii. Calcula si sigue en máquinas o pasa a 4.
 - iii. Actúa según el cálculo anterior
- 4. Si no va a máquina
 - a. **Comprueba si está siendo atendido.**
 - b. Si no lo está, calculamos el comportamiento del cliente (si se va por cansarse de esperar, si se va a máquinas) o si se va al baño y pierde su turno.
 - c. Si se fuera y **se escribe en el log**, se daría fin al hilo Cliente y **se liberaría el espacio en la cola**.
 - d. Si se va a máquinas pasaría a 3.
 - e. Sino debe dormir 3 segundos y vuelve a 4.
- 5. Si está siendo atendido por el recepcionista debemos esperar a que termine (puede comprobar cada 2 segundos).
- 6. Calculamos si va a coger los ascensores
 - a. Si va a cogerlos
 - i. **Cambia el valor de la variable ascensor del cliente**
 - ii. Intenta acceder al ascensor si no está funcionando
 - iii. Si está funcionando espera 3 segundos y vuelve a comprobar.
 - 1. **Guardamos el log en que está esperando por el ascensor**
 - iv. Sino
 - 1. Si aún no hay 6 espera (var condición)
 - 2. **Guardamos el log que está en el ascensor**
 - 3. Si es el 6, cierra y pone a funcionar el ascensor
 - a. Duerme
 - b. Deja el ascensor y permite que el siguiente dentro del ascensor actue
 - c. El último cierra la puerta y marca que el ascensor ha dejado de funcionar.
 - d. **Guardamos el log en que deja el ascensor**
 - b. Si no los coge
- 7. **Libera su posición en cola de clientes y se va.**
- 8. **Escribe en el log**
- 9. Fin del hilo Cliente.

}

AccionesRecepcionista{

1. **Buscar al cliente para atender de su tipo, esto es el que más tiempo lleve esperando.**
 - a. Si no hay clientes para atender espero un segundo y vuelvo a 1.
2. **Cambiamos el flag de atendido.**
3. Calculamos el tipo de atención y en función de esto el tiempo de atención (el 80%, 10%, 10%).
4. **Guardamos en el log que comienza la atención**
5. Dormimos el tiempo de atención.
6. **Guardamos en el log que finaliza la atención**
7. **Guardamos en el log el motivo del fin de la atención.**
8. **Cambiamos el flag de atendido**
9. Mira si le toca tomar café (solo en el caso de los no vip).
10. Volvemos al paso 1 y **buscamos el siguiente.**

```
}
```

Notas: las zonas coloreadas en rojo, azul, verde y morado son zonas de exclusión mutua que deben ser controladas. Para la parte morada se deben usar variables condición.

Escritura de mensajes en log: Es recomendable utilizar una función parecida a esta para evitar repetir líneas de código. Recibe como parámetros dos cadenas de caracteres, una para el identificador de vehículo o mecánico y otra para el mensaje (la fecha la calcula la propia función:

```
void writeLogMessage(char *id, char *msg) {
    // Calculamos la hora actual
    time_t now = time(0);
    struct tm *tlocal = localtime(&now);
    char stnow[25];
    strftime(stnow, 25, "%d/%m/%y %H:%M:%S", tlocal);
    // Escribimos en el log
    logFile = fopen(logFileName, "a");
    fprintf(logFile, "[%s] %s: %s\n", stnow, id, msg);
    fclose(logFile);
}
```