

DISEÑO DE LA PRÁCTICA FINAL

Declaraciones GLOBALES:

- Semáforos y variables condición
- Contador de usuarios
- Lista de 10 usuarios:
 - Id
 - Facturado
 - Atendido
 - Tipo
- UsuarioEnControl
- Fichero de log (FILE * logFile);

main {

1. signal o sigaction SIGUSR1, nuevoUsuario normal
2. signal o sigaction SIGUSR2, nuevoVip
3. Inicializar recursos (¡Ojo!, Inicializar!=Declarar).
 - a. Semáforos.
 - b. Contador de usuarios.
 - c. Lista de usuarios id 0, atendido 0, facturado 0.
 - d. Lista de facturadores.
 - e. Usuario en control
 - f. Fichero de Log
 - g. Variable condición
4. Crear 2 hilos facturadores.
5. Crear el hilo agente de control
6. Esperar señal SIGUSR1 o SIGUSR2 o señal de finalización
7. Esperar por señales de forma infinita.

}

nuevoUsuario{

1. Comprobar si hay espacio en la lista de facturación.
 - a. Si lo hay
 - i. Se añade el usuario.
 - ii. Contador de usuarios se incrementa.
 - iii. nuevoUsuario.id = ContadorUsuarios.
 - iv. nuevoUsuario.atendido=0
 - v. tipo=Depende de la señal recibida.
 - vi. Creamos hilo para el usuario.
 - b. Si no hay espacio
 - i. Se ignora la llamada.

}

AccionesUsuario{

1. Guardar en el log la hora de entrada.
2. Guardar en el log el tipo de usuario.
3. Duerme 4 segundos
4. Comprueba si está siendo atendido.
5. Si no lo está, calculamos el comportamiento del usuario (si se va o si se queda)
 - a. Si se va al baño o se cansa y se escribe en el log, se daría fin al hilo Usuario y se liberaría el espacio en la cola.
 - b. Sino debe dormir 4 segundos y vuelve a 4.
6. Si está siendo atendido por el usuario debemos esperar a que termine.
7. Cuando termine
 - a. Si ya ha sido facturado tiene que esperar por control
 - b. Si entra en control debe liberar la cola de facturación

- c. Se queda esperando a que pase la inspección y libera el control
- d. Cuando acaba libera el control
- e. Guardamos el log en que deja el control
- f. Imprime que embarca y se guarda en el log
- g. Libera control

8. Si no ha facturado
 - a. Libera su posición en cola de facturación y se va
 - b. Escribe en el log
9. Fin del hilo Usuario.

}

AccionesFacturador{

1. Buscar el primer vehículo para atender, esto es el que más tiempo lleve esperando.
 - a. Si no hay de mi tipo busco uno de la otra (si el usuario es normal y facturador es de tipo vip y la cola del otro tiene más de un usuario, lo atiende).
 - b. Si no hay usuarios para atender espero un segundo y vuelvo a 1.
2. Calculamos el tipo de facturación y en función de esto el tiempo de atención.
3. Cambiamos el flag de atendido.
4. Guardamos en el log la hora de atención.
5. Dormimos el tiempo de atención.
6. Guardamos en el log la hora de fin de la atención.
7. Guardamos en el log el motivo del fin de la atención.
8. Cambiamos el flag de atendido.
9. Incrementamos el contador de usuarios atendidos.
10. Mira si le toca tomar café.
11. Volvemos al paso 1 y buscamos el siguiente (siempre con prioridad a su tipo).

}

Acciones AgenteSeguridad{

1. Toma el mutex
2. Comprueba que haya algún usuario esperando por seguridad
3. Si no lo hay queda esperando a que llegue
4. Si hay alguno escribe en el log y calcula la atención
5. Según el tipo de atención duerme unos segundos u otros
6. Cuando termina de dormir escribe en el log
7. Avisa al usuario de que ha terminado la atención
8. Libera el mutex

}

Notas: las zonas coloreadas en rojo, azul y morado son zonas de exclusión mutua que deben ser controladas. Para la parte verde se recomienda el uso de 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[19];
    strftime(stnow, 19, "%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);
}
```