

Republica Bolivariana de Venezuela
Ministerio del Poder Popular para la Educacion Superior
Universidad Nacional Experimental de Guayana.
Vice-Rectorado Academico.
Proyecto de Carrera: Ingenieria Informatica
Asinatura: Sistemas Operativos

INFORME PROYECTO II

Docente:
Andres Caniumilla

Seccion :“1”

Integrantes:

Alonso Salas 20840287

Jesus Duran 24037328

Ciudad Guayana 9 de Marzo

Descripción de la Estructura de Datos Utilizada

tipo_tarjetas: es una estructura que simula los cartones de bingo, contienen los numeros y el id del participante.

```
struct tipo_tarjetas {  
    int numeros_tarjetas[NUMEROS];  
    int id_participante;  
}
```

memoria_compartida: es la estructura que va a estar cargada en memoria compartida, contiene datos sobre la partida actual, y en ella esta contenida la mayoria de la informacion para el juego.

```
typedef struct {  
    int idcantor;  
    int identificadores[MAX_PARTICIPANTES];           //estructura de la memoria compartida  
    int participantes;  
    int juez;  
    tipo_tarjetas tarjetas[MAX_PARTICIPANTES * MAX_TARJETAS_JUGADOR];  
    int ganador;  
    int numeros_cantados[MAX_BOLAS];  
} memoria_compartida;
```

Mi_tipo_Mensaje: es la estructura utilizada para el paso de mensajes.

```
typedef struct {  
    long Id_Mensaje;  
    int Dato_Numerico;           //estructura para el paso de mensajes  
    char Mensaje[20];  
} Mi_Tipo_Mensaje;
```

Tabla de Funciones Definidas:

En la clase cantor:

Publicas:

```
int crearMemoria();  
void crearColaMsj();  
void prepararTarjetas();  
int crearSemaforo();  
void cantarNumero();  
void ConectarMemoria();  
bool bomboVacio();  
void inicializarMemoria();
```

Globales:

```
int ultimoJugador();  
void AsignarTarjetas(int id);  
void manejador(int signum);  
void bloquearSemaforo(int id_sem);  
void desbloquearSemaforo(int id_sem);
```

En la clase participantes:

Publicos:

```
void Jugar();  
void cantarBingo();  
void comprobarBingo();  
void tacharNumero(int num);
```

```
void jugarmanual();
void jugarautomatico();
void imprimirtarjetas();
void manejador(int senal);
void otros_procesos();
void crearColaMsj();
bool IngresarSala();
```

Globales:

```
bool Bingo();
void manejador(int senal);
```

En la clase juez:

Publicos:

```
bool IngresarSala();
void ConectarMemoria();
void manejador(int sig);
void VerificarBingo();
bool Bingo();
```

Globales:

```
void manejador(int signum);
```

Diccionarios de variables:**Cantor:**

```
memoria_compartida *Memoria;
Mi_Tipo_Mensaje Mensaje; /
int memoria_global;
int Id_Cola_Mensajes;
int esperar = 0;
int semaforo;
cantor can;
```

Participante:

```
memoria_compartida *Memoria;
Mi_Tipo_Mensaje Mensaje;
int Id_Cola_Mensajes;
participante n;
int repetir;
```

Juez:

```
memoria_compartida *Memoria;
Juez n;
bool juego=true;
```

Especificaciones:

Entrada: no hay especificaciones de entrada.

Salida: El juego genera salida con respecto a conteo de numero, si el participante que canta bingo y su carton es comprobado, se le da como ganador.

Consideraciones:

Bueno profesor, implementamos los dos metodos de juego (manual y automatico), pero el metodo manual tiene problemas y funciona como deberia funcionar, corralo con el modo manual, que si funciona. Implementamos los conceptos que pudimos aprender en la materia (memoria compartida, cola de mensajes, alarmas, senales y semaforos).