

Sistemas Operativos

Trabajo Práctico 2019

Trenes



Profesor: Ing. Mostovoi Alejandro

Integrantes del Grupo:

Fernandez Gonzalo

Palacios Facundo

Pignatta Mauro

Pirrota Ezequiel

Ugobono Alejandro

Wiñar Mariano

Índice

Estructura de Datos	3
Estructura Trenes	3
Funciones y Aplicaciones	3
TREN inicializarTren(char *arch)	3
void partir(TREN tren)	3
void cargarCombustible(int *combustible)	3
void armarMensajeRegistrarse(TREN tren,char *mensaje)	4
void armarMensajeEstadoDelTren(TREN t, char * mensaje)	4
void armarMensajeExit(TREN tren, char * mensaje)	4
void armarMensajePartir(TREN tren ,char * mensaje)	4

Funciones de Tren

Realizan el recorrido de cada proceso llamado Tren a lo largo de todo el programa, pasa por toda la rama de comunicaciones entre el proceso tren y los procesos estación, con la característica de servidor/cliente, como así también las operaciones del mismo y los informes de su estado y condición.

1- Estructuras de Datos:

1.1- Estructura Trenes

```
typedef struct{
    int ID;
    int combustible;
    char modelo[10];
    char estOrigen[max_nombre_est];
    char estDestino[max_nombre_est];
    int tiempoRestante;
    int migrado;
    int nCliente;
}TREN;
```

2- Funciones y Aplicaciones:

2.1- TREN inicializarTren(char *arch):

Aplicación:

Inicializa el Tren con todas las variables,

Parámetros:

char * arch puntero a

Retorna:

Devuelve la variable tipo Tren con toda la info cargada.

int tipoAviso es 1 para avisar que viaja, 2 para avisar que llego.

2.2- void partir(TREN tren):

Aplicación:

Dibuja el Tren y resta el tiempo,

Parámetros:

TREN tren variable tipo tren.

2.3- void cargarCombustible(int *combustible):

Aplicación:

Carga Combustible al tren que lo solicite,

Parámetros:

int * combustible variable para indicar cantidad.

2.4- void armarMensajeRegistrarse(TREN tren,char *mensaje):

Aplicación:

Arma en una variable el mensaje con los datos del tren,

Parámetros:

TREN tren una variable del tipo TREN,

char * mensaje la variable del mensaje a enviar.

2.5- void armarMensajeEstadoDelTren(TREN t, char * mensaje):

Aplicación:

Arma el mensaje con el estado del tren,

Parámetros:

TREN tren una variable del tipo TREN,

char * mensaje la variable del mensaje a enviar.

2.6- void armarMensajeExit(TREN tren, char * mensaje):

Aplicación:

Arma el mensaje para salir del programa,

Parámetros:

TREN tren una variable del tipo TREN,

char * mensaje la variable del mensaje a enviar.

2.7- void armarMensajePartir(TREN tren ,char * mensaje):

Aplicación:

Arma el mensaje para indicarle a la estación que quiere partir el tren,

Parámetros:

TREN tren una variable del tipo TREN,

char * mensaje la variable del mensaje a enviar.