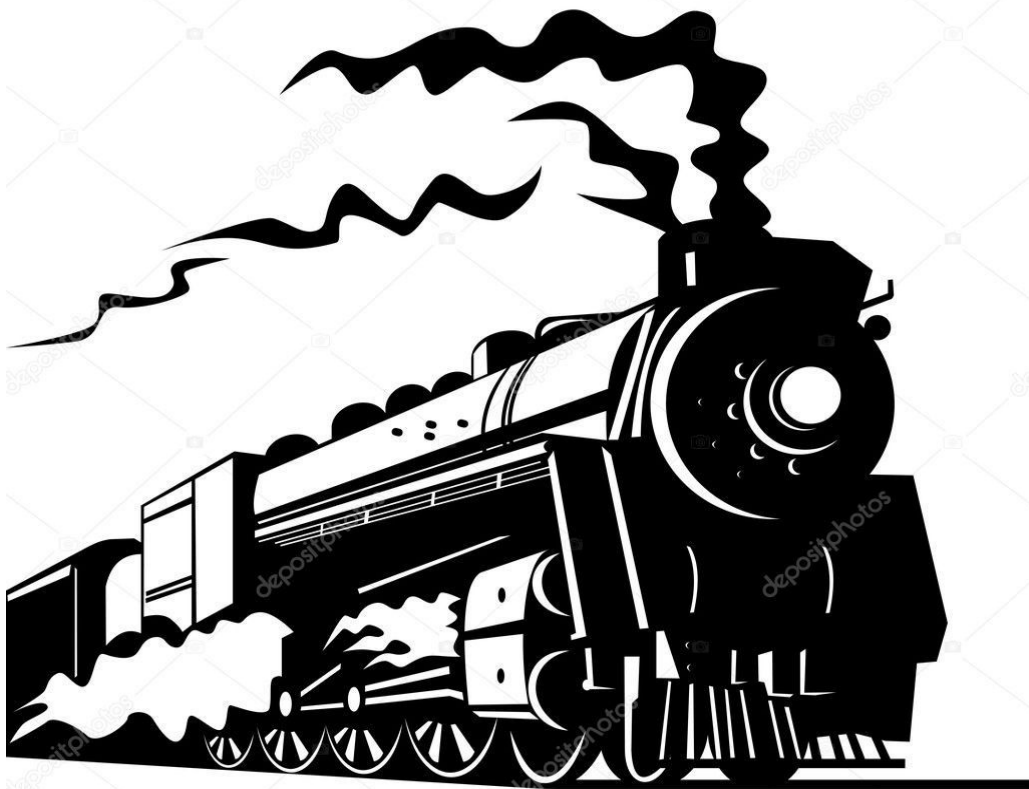


Sistemas Operativos

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
Interfaz Gráfica	3
Estructura del tipo enum para los colores a utilizar	3
Funciones y Aplicaciones	3
void initUserInterface(ST_APP_WINDOW *)	3
void drawUserInterface(ST_APP_WINDOW *)	3
void printWindowTitle(WINDOW *pWin, const char * message)	4
void printMessage(ST_APP_WINDOW *pWindow, const char * message, COLOUR colour)	4
void unInitUserInterface(ST_APP_WINDOW *)	4
void clearCmdWindow(WINDOW *pWin)	4
void clearLogWindow(WINDOW *pWin)	4
void printHelp(ST_APP_WINDOW *pAppWin)	4
void imprimirAndénAsignado(ST_APP_WINDOW *pWin)	5
void DibujarTrenViajando(WINDOW *pLogWindow, int * tiempoRestante)	5
void salirDelPrograma(TREN tren, int client, ST_APP_WINDOW * pWin)	5
void InterfazGrafica(void * argumentos)	5

Interfaz de Tren

Es la conexión funcional entre los dos procesos, en este caso los Trenes como Cliente y las Estaciones como Servidores de esos Clientes, es decir una relación servidor/cliente, proporcionándonos así la comunicación entre ambos, el envío de datos e información necesario para el funcionamiento total del programa, además de las funciones relacionadas al dibujo en pantalla a través de la interfaz gráfica con ncurses

1- Estructuras de Datos:

1.1- Interfaz Gráfica

```
typedef struct {  
    WINDOW *pAppFrame;  
    WINDOW *pLogFrame;  
    WINDOW *pLogWindow;  
    WINDOW *pCmdFrame;  
    WINDOW *pCmdWindow;  
} ST_APP_WINDOW;
```

1.2 Estructura del tipo enum para los colores a utilizar

```
typedef enum {RED = 1, GREEN, BLUE, WHITE, YELLOW, CYAN,} COLOUR;
```

2- Funciones y Aplicaciones:

2.1- void initUserInterface(ST_APP_WINDOW *)

Aplicación:

Crea las ventanas de la app. Asocia colores con las ventanas, Crea en pantalla un grupo de ventanas con la siguiente estructura:
Log: muestra mensajes,
Cmd: permite el ingreso de comandos al usuario.

Parámetros:

ST_APP_WINDOW * puntero a estructura que contiene las ventanas.

2.2 void drawUserInterface(ST_APP_WINDOW *):

Aplicación:

Dibuja el Marco de la App, Ventana Log, Ventana Cmd, Ventana Reg,

Parámetros:

ST_APP_WINDOW * puntero a estructura que contiene las ventanas.

2.3 void printWindowTitle(WINDOW *pWin, const char * message):

Aplicación:

Imprime en la pantalla tipo Log en la parte superior,

Parámetros:

- * pWindow puntero a la estructura que contiene las ventanas,
- * message contiene el mensaje a imprimir,

2.4 void printMessage(ST_APP_WINDOW *pWindow, const char * message, COLOUR colour):

Aplicación:

Imprime mensaje, param * pWindow puntero a la estructura que contiene las ventanas,

Parámetros:

- * message contiene el mensaje a imprimir,
- const char * colour de la estructura tipo enum con los colores a utilizar.

2.5 void unInitUserInterface(ST_APP_WINDOW *):

Aplicación:

Inicia el usuario de la interfaz grafica,

Parámetros:

- * variable global para la interfaz grafica.

2.6 void clearCmdWindow(WINDOW *pWin):

Aplicación:

Limpia la ventana de comandos,

Parámetros:

- * pWin variable global, ventana de comandos,

Retorna:

ERR_OK.

2.7 void clearLogWindow(WINDOW *pWin):

Aplicación:

Limpia la ventana de comandos y hace un refresco de pantalla,
Limpia la pantalla del Log,

Parámetros:

- * pWin variable global, ventana de comandos,

Retorna:

ERR_OK.

2.8 void printHelp(ST_APP_WINDOW *pAppWin):

Aplicación:

Imprime la Ayuda del programa,

Parámetros:

- * pAppWin representa la Variable global para la interfaz grafica.

2.9 void imprimirAndenAsignado(ST_APP_WINDOW *pWin):

Aplicación:

Imprime un mensaje que se le a asignado el andén al tren,

Parámetros:

* pAppWin representa la Variable global para la interfaz grafica.

2.10 void DibujarTrenViajando(WINDOW *pLogWindow, int * tiempoRestante):

Aplicación:

Muestra el Tren viajando,

Parámetros:

* pLogWindow representa la Variable global,

* puntero al tiempo restante del tren para actualizarlo.

2.11 void salirDelPrograma(TREN tren, int client, ST_APP_WINDOW * pWin):

Aplicación:

Salir del programa,

Parámetros:

TREN tren estructura de datos tren,

int client numero de cliente servidor,

* pAppWin representa la Variable global para la interfaz grafica.

2.12 void InterfazGrafica(void * argumentos):

Aplicación:

Para el hilo que se encarga de la interfaz grafica,

Parámetros:

* argumentos.