

## Ejercicio Clase 28-10-2019

### Parte Teórica

1. ¿En qué casos utilizaría librerías dinámicas? Justifique.
2. La función `fork()` se utiliza para crear un nuevo proceso. ¿Qué significado tiene el valor que retorna?
3. Usted corre el comando

```
$ ps a
PID   TTY   STAT TIME   COMMAND
16347 pts/6  Ss    0:00   bash
16416 pts/6  R+    0:00   ps a
```

¿Qué significa que el proceso de ejecución de éste comando esté en estado R+ al ejecutarse el mismo?

### Parte Práctica

Para un sistema de control de tránsito en un cruce se desea registrar el paso de los vehículos durante un tiempo determinado. Se recibe por una fifo llamada “./trama” la siguiente trama de datos:

Header 0x5A	Patente	Tipo	Direccion	CRC	Trailer 0xA5
1 byte	6 bytes	1 byte	1 byte	4 bytes	1 byte

- Header y Trailer indican comienzo y final de la trama
- Tipo es un valor numérico que indica el tipo de vehículo que pasó por el control de tránsito, siempre mayor a cero.
- Dirección indica la dirección del vehículo (N=Norte, O=Oeste, S=Sur, E=Este)
- CRC es un campo de detección de errores (tipo int) para comprobar que la trama ha sido recibida de forma íntegra.

La señal SIGUSR1 nos informa que la fifo contiene datos para recibir. La fifo puede contener una o más tramas cada vez que se recibe la señal, deben leerse todas las que haya disponibles hasta que la fifo quede vacía.

Los campos de la trama son de ancho fijo y no contienen terminadores en ningún caso.

Todos los pasos por el cruce se almacenarán en una lista simplemente enlazada utilizando la siguiente estructura:

```
typedef struct nodo_s {  
    time_t timestamp;  
    char patente[10];  
    unsigned short int tipo;  
    char direccion;  
    struct nodo_s * next;  
} nodo_t;
```

### Se pide implementar las siguientes funciones:

*int recibir\_trama(char \* buff);*

Lee la trama de la named fifo especificada, y carga su contenido debidamente terminado en el buffer recibido como parámetro. Debe verificar la trama antes de considerarla válida. Retorna OK en caso de éxito, o ERROR en caso de no poder leer la trama completa, o que la validación falle.

*int validar\_trama(char \* buff);*

Recibe una trama en buff y realiza las siguientes comprobaciones. Todas ellas deben resultar exitosas, o la trama se considerará inválida:

- Header y el Trailer deben contener sus valores respectivos.
- Patente debe contener sólo letras mayúsculas en los primeros tres caracteres y sólo números en los últimos 3.
- Tipo debe ser siempre mayor a cero y estar dentro de los valores posibles.
- Dirección debe ser solamente N, O, S o E.
- El CRC debe verificarse con la siguiente función, que ya está implementada y no forma parte del ejercicio.

*int crc32(int crc, char \* buff);*

*int registrar\_paso(nodo\_t \*\*L, char \* trama)*

Recibe una trama considerada válida y la agrega en la lista simplemente enlazada, siempre al final. Debe setear el timestamp del nodo al momento de insertarse.

*void signal\_paso(int signum);*

Recibe las señales SIGUSR1 y SIGALRM y actúa en consecuencia. Debe considerar que en esta función es donde se define comienzo de la lista, es invocada asincrónicamente en cualquier momento, y no recibe ningún otro parámetro. Está prohibido el uso de variables globales. Indique cómo registraría esta función para ser invocada ante las diferentes señales.

*char \* tipo\_str(char tipo);*

Recibe el valor numérico del tipo de vehículo y devuelve el nombre correspondiente. Su valores posibles son:

- 1 = Motovehiculo
- 2 = Auto
- 3 = Utilitario
- 4 = Camioneta
- 5 = Camión sin acoplado
- 6 = Camión con acoplado
- En caso de recibir un valor incorrecto, devuelve NULL.

*int exportar\_reporte(nodo\_t \* L);*

Mediante la señal SIGALRM , configurada para dispararse cada 30 segundos, se exportará un reporte en un archivo "cruce.csv" de la cantidad de cruces por tipo de vehículo, con el siguiente formato:

Tipo, Cantidad Norte, Cantidad Oeste, Cantidad Sur, Cantidad Este