

Práctica 0
Práctica Introductoria

1. Introducción y Funcionamiento

Se realizará una aplicación en C sobre QNX 6.3 que controla un conjunto de vehículos simulados. Los vehículos detectan colisiones y cambian de dirección cada vez que sufren una de ellas. Los vehículos desaparecen cuando han sufrido un cierto número de colisiones. Cuando todos los vehículos acaban, termina la aplicación.

Para el desarrollo de este ejercicio se proporcionará a los alumnos el código C de los diferentes módulos de la aplicación, con funcionalidad limitada. Es responsabilidad del alumno modificar y completar el código fuente y compilarlo para obtener un comportamiento satisfactorio.

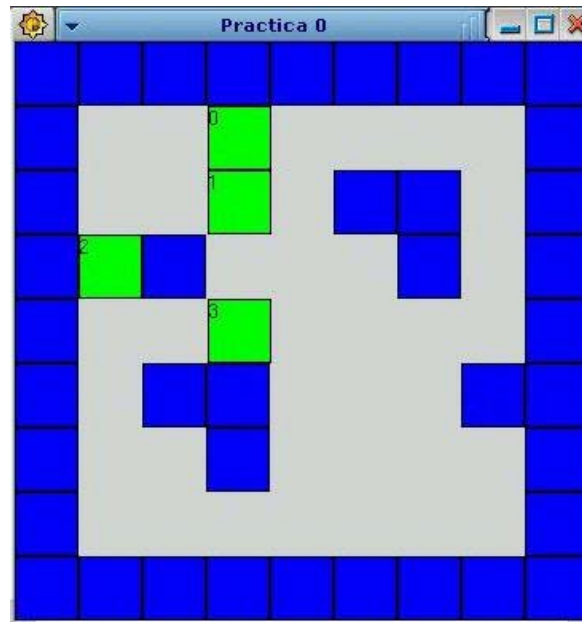


Figura 1: Representación gráfica

Arquitectura software

La aplicación está formada por los procesos:

- **pr0**: El ejecutable **pr0** se compone de los módulos **practica0.c** (a modificar por el alumno) y **api_grafi.o** (interfaz de gráficos). Este proceso se encarga de inicializar la interfaz gráfica y de crear los procesos de control de vehículos. Finalmente espera a que todos ellos acaben y cierra la interfaz gráfica. La inicialización y cierre de los gráficos se realiza mediante un proceso hijo, como se explica más adelante.
- Procesos de control de vehículos. Todos comparten el mismo ejecutable **veh0**.
- El proceso **grafi**, que actúa como servidor de gráficos. Las funciones de **grafi** se utilizan por medio de **api_grafi.o**.

El alumno sólo debe modificar el fichero **practica0.c**.

Funcionamiento de los procesos de control de vehículos:

El ejecutable **veh0** se encarga de crear un vehículo en la ventana gráfica de nombre “Practica 0” y de moverlo de modo que cambia de dirección cada vez que choca con un obstáculo. Cuando se ha producido un número máximo de choques el vehículo desaparece y el proceso acaba.

- **Argumentos de la línea de comandos:** veh0 utiliza los argumentos 1 a 4, si existen; si no existe un argumento determinado utiliza un valor por defecto. El significado de cada argumento es el siguiente:
 - Argumento 1: Entero; índice del vehículo que aparece en la figura. Valor por defecto: 0.
 - Argumento 2: Entero, fila inicial. Valor por defecto: 1.
 - Argumento 3: Entero, columna inicial. Valor por defecto: 1.
 - Argumento 4: Entero, número de máximo de choques antes de salir. Valor por defecto: 10.
- **Movimiento del vehículo:** El vehículo se mueve inicialmente hacia la derecha. Si se encuentra con un obstáculo espera medio segundo y cambia de dirección cíclicamente: abajo, izquierda, arriba y otra vez derecha para cerrar el ciclo.
- **Resultado de terminación:** El proceso de vehículo entrega su índice como resultado de terminación.

Explicación del arranque del sistema de soporte gráfico:

El arranque y la parada de los gráficos se realizan mediante la función **crea_grafi** ejecutada en un proceso separado. De este modo el proceso padre **pr0** podría crear procesos de control de vehículos después de existir la interfaz gráfica (no es posible en QNX ejecutar **fork** en un proceso después de utilizar **abrir_graficos**, pues esta función crea internamente varios hilos).

2. Trabajo del alumno

El trabajo a realizar consta de dos partes:

- **Parte obligatoria:** A realizar en el laboratorio en horario de prácticas. Consiste en modificar el código que se proporciona para conseguir la funcionalidad descrita en los puntos anteriores.
- **Parte voluntaria:** El alumno podrá realizar mejoras y desarrollos adicionales de la práctica, que influirán positivamente en la calificación final. Como ejemplo se propone las siguientes modificaciones:
 - Modificar los parámetros de funcionamiento (número de vehículos, posición inicial de los vehículos, número máximo de choques...) mediante los argumentos de la línea de comandos de **pr0**.
 - Modificar **pr0** de manera que vuelva a crear los vehículos a medida que van desapareciendo.
 - Escribir un programa para implementar los procesos de control de vehiculos.

Deberá entregarse la siguiente documentación:

- Memoria del trabajo realizado, incluyendo modificaciones, junto con una explicación detallada del funcionamiento del programa. **No se trata de repetir el enunciado.**
- Ficheros fuentes, de manera que pueda comprobarse el funcionamiento de la aplicación.

3. Ficheros disponibles

- **practica0.c**: Fichero fuente para el proceso **vehículo** (a ampliar y modificar).
- **practica0**: Fichero ejecutable para los procesos **vehículo y arranque**.
- **practica0.h**: Cabecera común para los programas de la práctica 1.
- **api_grafi.o**: Fichero objeto con las funciones de interfaz para el servidor de gráficos.
- **api_grafi.h**: Cabecera para las funciones de **api_grafi.o**.
- **grafi**: Ejecutable del servidor de gráficos.
- **veh0**: Ejecutable de los procesos de control de vehículos.
- **ej_referencia/pr0**: Ejecutable de **pr0** correcto para servir de referencia.

Para compilar el ejecutable **práctica0** pueden utilizarse los siguientes comandos en QNX 6.3.2:

```
gcc -o pr0 practica0.c api_grafi.o
```

En Ubuntu el comando es algo distinto:

```
gcc -o pr0 practica0.c api_grafi.o -lrt -lpthread
```

4. Cabeceras

4.1. Cabecera practica0.h

```
/* Cabecera para practica 0 GIERM (10/2015) */
/* Copyright (C) Joaquin Ferruz Melero 2015 */

#ifndef __PRACTICA0_H__
#define __PRACTICA0_H__

/* Acceso a interfaz grafica */

#include "api_grafi.h"

/* Dimensiones del dibujo */

#define ANCHO 9                /* Ancho del mapa */
#define ALTO 9                 /* Alto del mapa */
#define ESCALA_DIB 40          /* Escala */
#define N_VEH 4                /* Numero de vehiculos */

/* Mapa:
   C: camino
   P: Pared */

#define P PARED                /* Definido en api_grafi.h */
#define C PASILLO              /* " " " " */

int mapa[ALTO][ANCHO] = {
    {P, P, P, P, P, P, P, P, P},
    {P, C, C, C, C, C, C, C, P},
    {P, C, C, C, C, P, P, C, P},
    {P, C, P, C, C, C, P, C, P},
    {P, C, C, C, C, C, C, C, P},
    {P, C, P, P, C, C, C, P, P},
    {P, C, C, P, C, C, C, C, P},
    {P, C, C, C, C, C, C, C, P},
}
```

```
                {P, P, P, P, P, P, P, P, P}  
            };  
#endif
```