

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
SISTEMAS OPERATIVOS 1



Práctica 2
MANUAL TÉCNICO

NOMBRE: WILLIAM FERNANDO VALLADARES MUÑOZ
CARNET: 201314510
FECHA: 18/09/2016

Índice

I.	Introducción.....	1
II.	Objetivos.....	2
III.	Práctica 2	
	a) Descripción de la Solución.....	3
	b) Organización de la Solución.....	4
	c) IDE y Sistema Operativo.....	4
	d) Librerías Externas Utilizadas.....	4
	e) Código Fuente.....	4
IV.	Contacto.....	5

I. Introducción

La presente aplicación consiste en una variante del juego “SPACE INVADERS”, utilizando el lenguaje C. Dicha variante, permite jugar a dos personas (en modo versus) utilizando instancias diferentes del programa, lo que permite jugar tanto de manera local, como remota utilizando conexiones SSH.

El juego cuenta con un sistema de puntaje, que ayuda a determinar que jugador es el ganador de la batalla. La aplicación se comunica con su otra instancia por medio de memoria compartida, haciendo uso de la versión 5 del algoritmo de Dekker y además de semáforos en su ejecución.

II. Objetivos

General

- Aprender sobre el manejo de la memoria compartida y así mismo sobre la aplicación de medidas antibloqueo entre procesos.

Específicos

- Aplicar el algoritmo de Dekker, dentro de una aplicación real.
- Utilizar la memoria compartida, para compartir valores entre las instancias del juego.
- Hacer uso de los semáforos del sistema operativo Linux.
- Aplicar dentro de un programa, los algoritmos vistos en el laboratorio.

III. Práctica 2

a) Descripción de la Solución

La aplicación se divide en 4 partes de código fuente:

- Main.c
Aquí se declara el punto de partida para la ejecución de la aplicación.
- Frame.c
Aquí se declaran las distintas pantallas que componen la aplicación:
 - void pantalla_bienvenida()
 - void pantalla_menu()
 - void print_menu(WINDOW *menu_win, int highlight)
 - void pantalla_espera()
 - void pantalla_juego()
 - void pantalla_fin()
- Shared.c
Aquí se declaran los segmentos de memoria compartida y así mismo los mecanismos para evitar bloqueos al hacer uso de la región crítica:
 - void start_share()
 - void free_share()
 - void inicializar_semaforo()
 - void eliminar_semaforo()
 - void up_semaforo()
 - void down_semaforo()
 - int get_semaforo()
 - void pre_region_critica()
 - void post_region_critica()
- Game.c
Aquí se desarrolla la lógica del negocio de la aplicación, comunicándose constantemente con lo declarado en frame.c y en share.c:
 - void start_game(WINDOW* w1, WINDOW* w2, WINDOW* w4)
 - void *print_nave(void *win)
 - void actualizar_disparos()
 - void print_fila (int x, int y)
 - void actualizar_filas ()
 - void check_colision ()

b) Organización de la Solución

- Frame.c
- Game.c
- Main.c
- Shared.c
- Space_Invaders.pro
- Space_Invaders.pro.user

c) IDES y Sistema Operativo

El sistema operativo en que se realizó dicho proyecto es Manjaro Linux 16.08 y el IDE utilizado fue Qt Creator 4.1.0.

d) Librerías Externas Utilizadas

En el presente proyecto de laboratorio se emplearon las siguientes librerías externas:

- NCurses.h
- Pthread.h
- Time.h
- String.h
- Stdio.h
- Stdlib.h
- Sys/ipc.h
- Sys/sem.h
- Sys/types.h
- Sys/shm.h

e) Código Fuente

Para mayor análisis del código fuente del proyecto, dirigirse al enlace del repositorio de GitHub:

https://github.com/will9494/Space_Invaders

IV. Contacto

Nombre: William Fernando Valladares Muñoz

Ocupación: Estudiante Ingeniería en Sistemas en USAC

Correo: willfer9494@hotmail.com