

# Universidad Nacional Autónoma de México

## Facultad de Ciencias

Sistemas Operativos - Practica 1

Agosto de 2018

### Descripción de programas

1. Ordenamiento burbuja con apuntadores.
  - Realizar un programa en C, en el cual con cada iteración del ordenamiento burbuja, se modifiquen las referencias a los datos en memoria, y no los datos contenidos en la misma.
  - Se deberá imprimir en cada iteración las direcciones de memoria empleando formateadores de cadena específicos para esta acción (%p)
2. Keylogger
  - Empleando la llamada al sistema read en un sistema operativo derivado de UNIX, interceptar la información del dispositivo de entrada "teclado" y guardarla en un archivo de texto.
  - Las entradas del teclado se almacenan en estructuras(structs) de tipo input\_event, que almacenan entre otros datos el código de la tecla pulsada o liberada, la definición de este tipo de estructuras se encuentra en sys/types.h y linux/input.h.
  - El dispositivo que genera este tipo de estructuras se encuentra de manera lógica en /dev/input/ en sistemas derivados de UNIX y se maneja como un evento, el evento cambia entre cada sistema operativo.
3. Multiplicación dinámica de matrices.
  - Empleando las funciones malloc(), calloc(), realloc() y free(), crear arreglos bidimensionales que representen matrices a multiplicar.
  - Las dimensiones de las matrices deberán ser especificadas por el usuario, y no se debe emplear más espacio de almacenamiento que el requerido por las matrices.

- Se tiene la opción de solicitar los elementos de las matrices al usuario o generarlos de manera automática a través de funciones de generación de números aleatorios propias de C. En ambos casos se deben imprimir las matrices y el resultado de la multiplicación.

#### 4. 2K38Y

- Imprimir en pantalla la fecha exacta con segundos para la Ciudad de México en la que la forma de medir el tiempo con epoch desbordará el espacio de almacenamiento válido para un entero de 32 bits.
- Restricción. no más de 5 líneas de código dentro de la función main.

#### 5. (EXTRA 1) Borrado (in)seguro

- Empleando la llamada al sistema `memset()` implementar el borrado seguro de un archivo especificado a través de línea de comandos.
- Se debe eliminar el archivo tanto del sistema de archivos como de la memoria.
- En el espacio de memoria del archivo, se deben escribir ceros o información aleatoria.

#### 6. (EXTRA 2) Pseudoaleatorio

- Empleando el dispositivo `/dev/random` o `/dev/urandom` y la llamada al sistema `read()`, crear un generador de números aleatorios.
- Se deben leer la cantidad suficiente de bytes del dispositivo y almacenarlos en una variable de tipo `unsigned int` e imprimir su contenido.

## NOTAS

- Todos los programas deben realizarse en el lenguaje de programación C, y deben ser compilados a través de un Makefile con la bandera `Wall`
- Cada programa deberá estar debidamente documentado con imágenes y pruebas de ejecución en un archivo `README.md`
- En el directorio raíz deberán estar especificados los nombres de los integrantes del equipo que participan en la práctica.