

# INFORME TRABAJO PRÁCTICO DE ARQUITECTURA DE COMPUTADORAS 2DO CUATRIMESTRE

## **Integrantes:**

Juan Bensadón 57193

Guido Princ 57334

Año: 2017

## **General:**

Se reutilizaron los archivos provistos por la cátedra para la realización de este trabajo.

Al comienzo se trabajó en modo texto, realizando cada funcionalidad del proyecto orientada para que corra en dicha modalidad (excluyendo la función graficadora). Suponíamos que iba a resultar simple hacer el cambio entre el modo video y el modo texto para poder correr en cada uno las funciones correspondientes, pero al final (al realizar consulta con los profesores y compañeros) fuimos directamente por implementar todas las funciones en modo video para ahorrar tiempo ya que no contabamos con lo necesario para realizar lo que queríamos.

## **Realización del proyecto:**

Se utilizaron los conocimientos provistos por la cátedra, y se realizaron consultas tanto a los profesores como a nuestros compañeros cuando nos surgía una problemática o cuando surgían errores inesperados.

## **Interrupciones:**

Se agregaron las interrupciones de teclado y la int 80h para la realización del proyecto.

## **División:**

La librería provista por la cátedra ya favorecía la división entre el kernel y el user space, por lo que solo fue necesario implementar lo necesario en cada uno para cada funcionalidad del trabajo.

## **Intérprete de comandos:**

Se utiliza la función scanf y las interrupciones de teclado para determinar el carácter ingresado, y éste se guarda en un buffer para interpretar lo que el usuario escribió. Luego con un switch se verifica que función es requerida si cumple con los caracteres correctos y sus parámetros.

## **Funciones:**

- La función del tiempo fue tomada del ejercicio 3 de la práctica 5, y solo fue necesario implementarlo con printf para mostrar la misma.
- Las función printf funciona similar a las funciones provistas por la cátedra, pero fue modificada usando las system calls requeridas, e imprimiendo los caracteres correctos con la matriz otorgada por la cátedra para las letras, números, etc.
- La función scanf utiliza las syscalls y guarda en un buffer los caracteres ingresados, así facilita la ejecución de las otras funciones.
- El help espera el nombre de una función válida para mostrar su implementación.
- La graficadora limpia la pantalla, dibuja el cross axis y según los valores ingresados colorea los lugares correspondientes a la función ingresada (además colorea los de alrededor para que tenga más grosor la línea). Para la escala se toma como default que 1 pixel es 1 unidad.

- Se pueden probar las excepciones de zerodivision y overflow. Por razones que desconocemos, tuvimos que comentar la función para cuando ocurre la excepción de operación inválida ya que aparece una repetición en loop de la misma.

**Errores:**

Como ya se ha dicho, comentamos la función de la excepción de operación inválida por un error que no se pudo comprender, además no se pudieron agregar otras funciones que queríamos porque al agregar más funciones en la comparación de lo escrito en el buffer, ocurría el mismo error. Suponemos que debe ser algo del makefile en userland y con el sampleModule propio. Si se corre en la máquina virtual subida al campus funciona sin problema.

**A agregar:**

Por lo errores ya mencionados, no fueron agregados otras funciones que teníamos el interés de agregar. Tales son como el clear (para limpiar la pantalla, que ya está la función), que el help acepte el parámetro a pedir la ayuda, poder escalar la función graficadora y agregarle validación de números con punto flotante, entre otros.

Desde ya ante cualquier duda estamos a su disposición y pueden mandar un mail a [gprinc@itba.edu.ar](mailto:gprinc@itba.edu.ar) para responder cualquiera de sus consultas.