



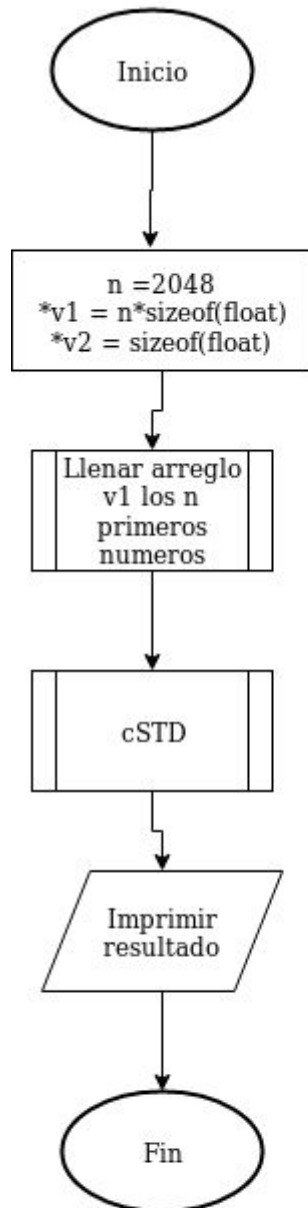
TAREA LABORATORIO N°2

Nombre: Antony Palacios Coronado
Código: 20141735

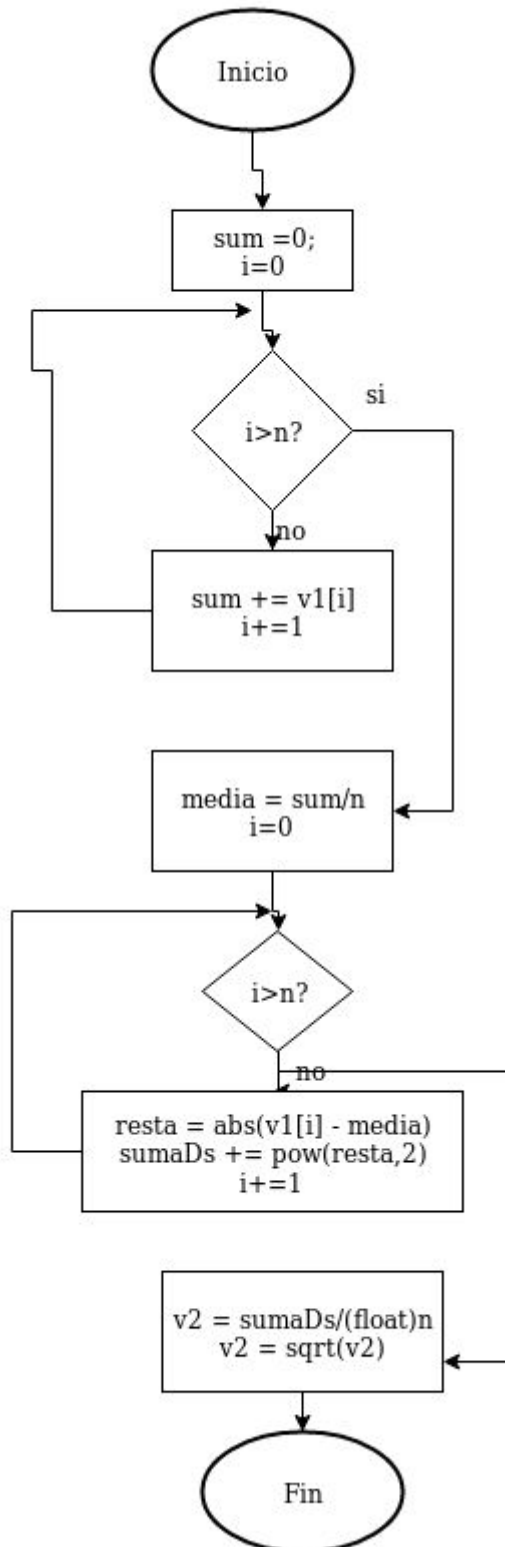
Curso: Organización y Arquitectura de Computadoras
Profesor: Stefano Romero
Horario: 0681

Diagrama de Flujo

Flujo Principal



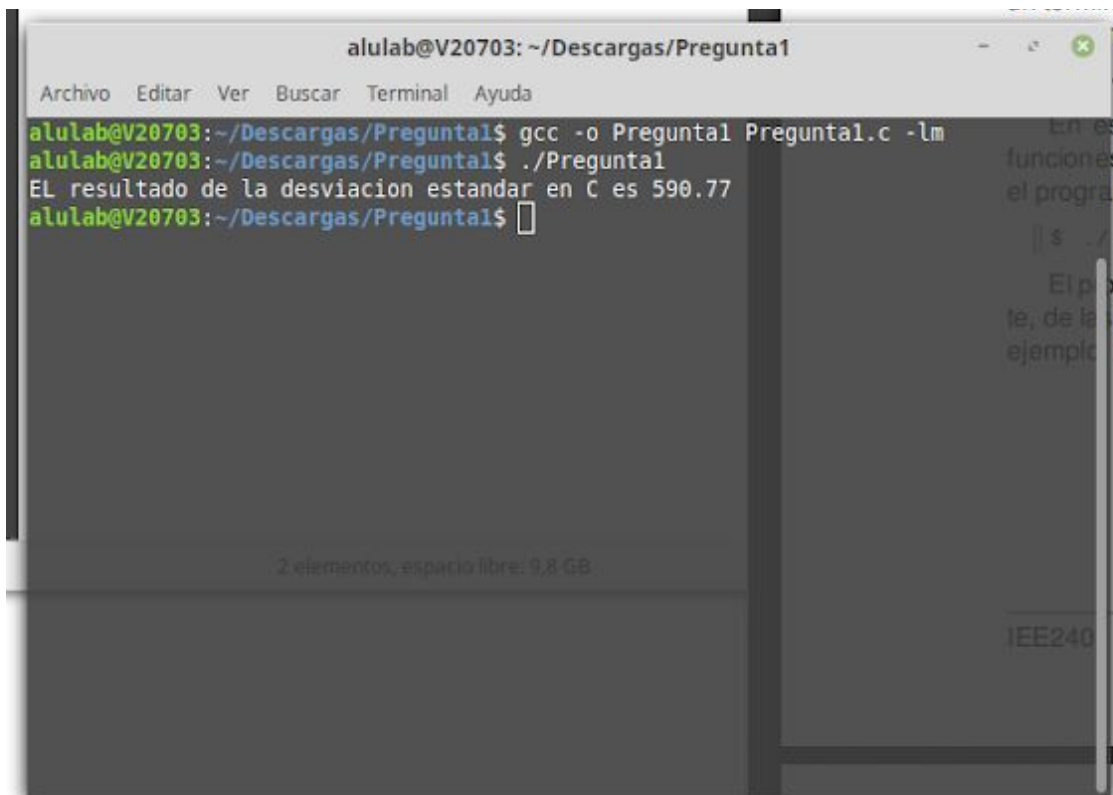
Flujo del subProceso cSTD



Descripción del Programa

Pregunta 1

El programa calcula la desviación de un arreglo de números en punto flotante, en este caso los 2048 primeros números. Primero se definen las variables y luego lleno el arreglo de 4 en 4. Luego se manda este arreglo a la función cSTD para evaluarlo. Se calcula primero la media, luego recorremos el arreglo nuevamente para restar cada elemento con la media, y esta resta se eleva a potencia 2 y se acumula en un sumador. Finalmente se divide esta suma con la cantidad N indicada y se calcula la raíz cuadrada.



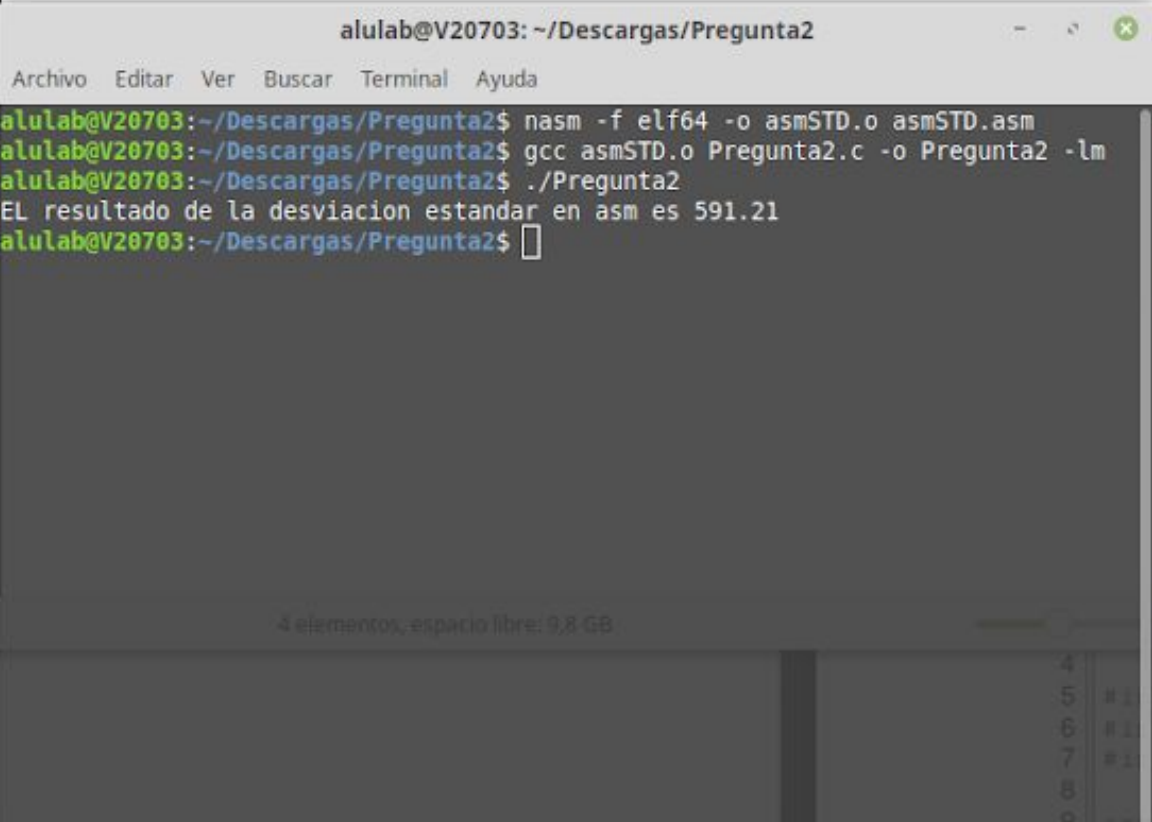
```
alulab@V20703: ~/Descargas/Pregunta1
Archivo Editar Ver Buscar Terminal Ayuda
alulab@V20703:~/Descargas/Pregunta1$ gcc -o Pregunta1 Pregunta1.c -lm
alulab@V20703:~/Descargas/Pregunta1$ ./Pregunta1
EL resultado de la desviacion estandar en C es 590.77
alulab@V20703:~/Descargas/Pregunta1$
```

2 elementos, espacio libre: 9.8 GB

IEE240

Pregunta 2

El programa asmSTD calcula primero la media sumando cada elemento del registro rdi y se acumula en xmm1. Luego se avanza 4 bytes el registro y se disminuye en 1 la cantidad N. Este proceso se repite hasta que N sea cero. Se divide xmm1 entre xmm3, el cuál contiene el número N en tipo float. Se regresa el puntero del registro al inicio para poder calcular la varianza, similar al proceso anteriormente descrito. Una vez obtenida la varianza en xmm2, se calcula su raíz cuadrada y el resultado se guarda en rdx, el cual retorna el resultado. Este programa es llamado por un programa similar al programa 1.



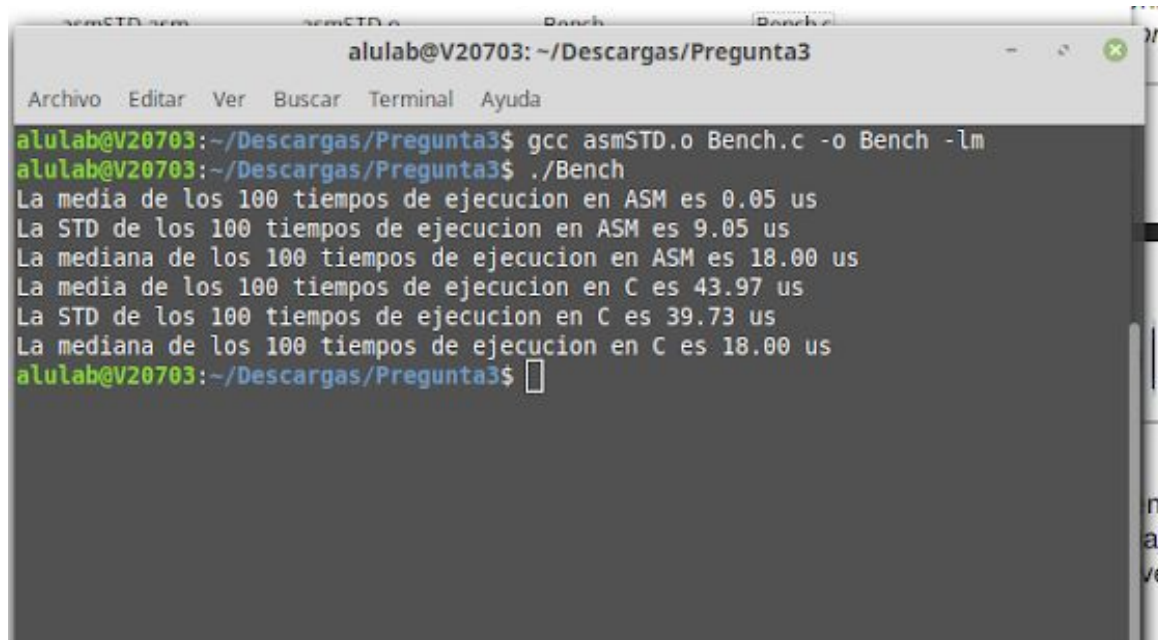
```
alulab@V20703: ~/Descargas/Pregunta2
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
alulab@V20703:~/Descargas/Pregunta2$ nasm -f elf64 -o asmSTD.o asmSTD.asm
alulab@V20703:~/Descargas/Pregunta2$ gcc asmSTD.o Pregunta2.c -o Pregunta2 -lm
alulab@V20703:~/Descargas/Pregunta2$ ./Pregunta2
EL resultado de la desviacion estandar en asm es 591.21
alulab@V20703:~/Descargas/Pregunta2$
```

4 elementos, espacio libre: 9,8 GB

| | |
|---|-----|
| 4 | 8.1 |
| 5 | 8.1 |
| 6 | 8.1 |
| 7 | 8.1 |
| 8 | 8.1 |
| 9 | 8.1 |

Pregunta 3

El programa utiliza las funciones cSTD y asmSTD para comparar los tiempos de ejecución en 100 iteraciones.



The screenshot shows a terminal window titled "alulab@V20703: ~/Descargas/Pregunta3". The window has a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The terminal content shows the following commands and output:

```
alulab@V20703:~/Descargas/Pregunta3$ gcc asmSTD.o Bench.c -o Bench -lm
alulab@V20703:~/Descargas/Pregunta3$ ./Bench
La media de los 100 tiempos de ejecucion en ASM es 0.05 us
La STD de los 100 tiempos de ejecucion en ASM es 9.05 us
La mediana de los 100 tiempos de ejecucion en ASM es 18.00 us
La media de los 100 tiempos de ejecucion en C es 43.97 us
La STD de los 100 tiempos de ejecucion en C es 39.73 us
La mediana de los 100 tiempos de ejecucion en C es 18.00 us
alulab@V20703:~/Descargas/Pregunta3$
```