

Informe Taller 2

Taller de Sistemas Operativos
Escuela de Ingeniería Informática

Abel Núñez Cataldo

Abel.nunez@alumnos.uv.cl

Resumen. *Este informe documenta los procedimientos que se llevaron a cabo sobre el desarrollo de un programa utilizando threads, y el diseño de una solución que permita el llenado de un arreglo y la suma de sus contenidos en forma paralela, usando las herramientas proveídas en clase.*

1. Introducción

Bash es una herramienta de desarrollo de software que trabaja directamente sobre el sistema operativo, agilizando la utilización de memoria y procesamiento de código. Para llevar a cabo este ejercicio se utilizó, el software de virtualización distribuido por Oracle llamado VirtualBox, con el cuál se instaló una máquina virtual de Ubuntu Server versión 18.04, que se basa principalmente en el uso de comandos Linux. En esta entrega se prioriza la implementación de threads (hilos), para lo cual se utilizara principalmente el lenguaje de programación C++ estándar 2014 o superior.

El objetivo de este taller, consiste en realizar el diseño e implementación de un programa que llene un arreglo de números enteros y luego los sume. Estas acciones deben ser ambas realizadas de forma paralela, utilizando threads POSIX.

Este documento cuenta con una introducción que abarca la tecnología utilizada, los objetivos a alcanzar. Seguida de la descripción del problema planteado en el taller, que incluye la descripción de los datos utilizados, el significado de las variables utilizadas y algunos ejemplos en que se instancian. Finalizando con el diseño de la solución, en la que se muestran diagramas de alto nivel y los métodos utilizados para la solución del problema.

2. Descripción del problema

El problema planteado es crear un programa que esté compuesto de dos módulos. Uno que llene un arreglo de números enteros aleatorios en forma paralela y otro que sume el contenido del arreglo en forma paralela. Se deben realizar pruebas de desempeño, que generen datos que permitan visualizar el tiempo de ejecución de ambos módulos, dependiendo del tamaño del problema y de la cantidad de threads utilizados. Se establece que la forma de ejecutar este programa debe ser la siguiente:

```
./sumArray -N <nro> -t <nro> -l <nro> -L <nro> [-h]
```

Parámetros:

-N	: tamaño del arreglo.
-t	: número de threads.
-l	: límite inferior rango aleatorio.
-L	: límite superior rango aleatorio
[-h]	: muestra la ayuda de uso y termina.

2.1 Descripción de los datos

Para esta situación a diferencia del taller anterior, no se hará uso de datos almacenados previamente en archivos de texto, sino que se utilizarán datos de tipo entero generados aleatoriamente para llenar un arreglo. Los otros datos que se apreciarán, son los del desempeño que tengan los procesos del programa a la hora de su ejecución.

3. Diseño de la solución

3.1 Metodología

La metodología consiste en analizar los requerimientos, diseñar un modelo del funcionamiento del script, donde se pueda ver un orden de los procesos a realizar para lograr las acciones de llenado del arreglo y la suma de su contenido. Una vez estudiado el funcionamiento general de la solución, se debe crear un modelo de diseño más específico para los módulos del taller, es decir, para cada módulo es necesario realizar un diseño de comportamiento.

3.2 Diseño

Para el diseño de la solución primero se procede a analizar el problema, estableciendo los siguientes requerimientos:

- Debe recibir como parámetros de entrada, el tamaño del arreglo, número de threads y rango de los números en el arreglo
- Generar un arreglo de tamaño dinámico vacío.
- Llenar el arreglo con números enteros aleatorios en forma paralela.
- Sumar el contenido del arreglo y almacenarlo.

Seguido por el diseño de un diagrama de alto nivel, que ayude a explicar el funcionamiento general de la posible solución para el problema. Para esto se creó un diseño basado en el ejemplo de llenado paralelo de arreglo proveído por el profesor, en el que se asume que el tamaño del arreglo es dinámico, configurable por parámetros de entrada. La solución se estructura en las etapas denominadas “*Etapas de llenado*” y “*Etapas de sumado*” (Ver Figura 1), esta última etapa se comporta de forma similar a la lógica de la primera.

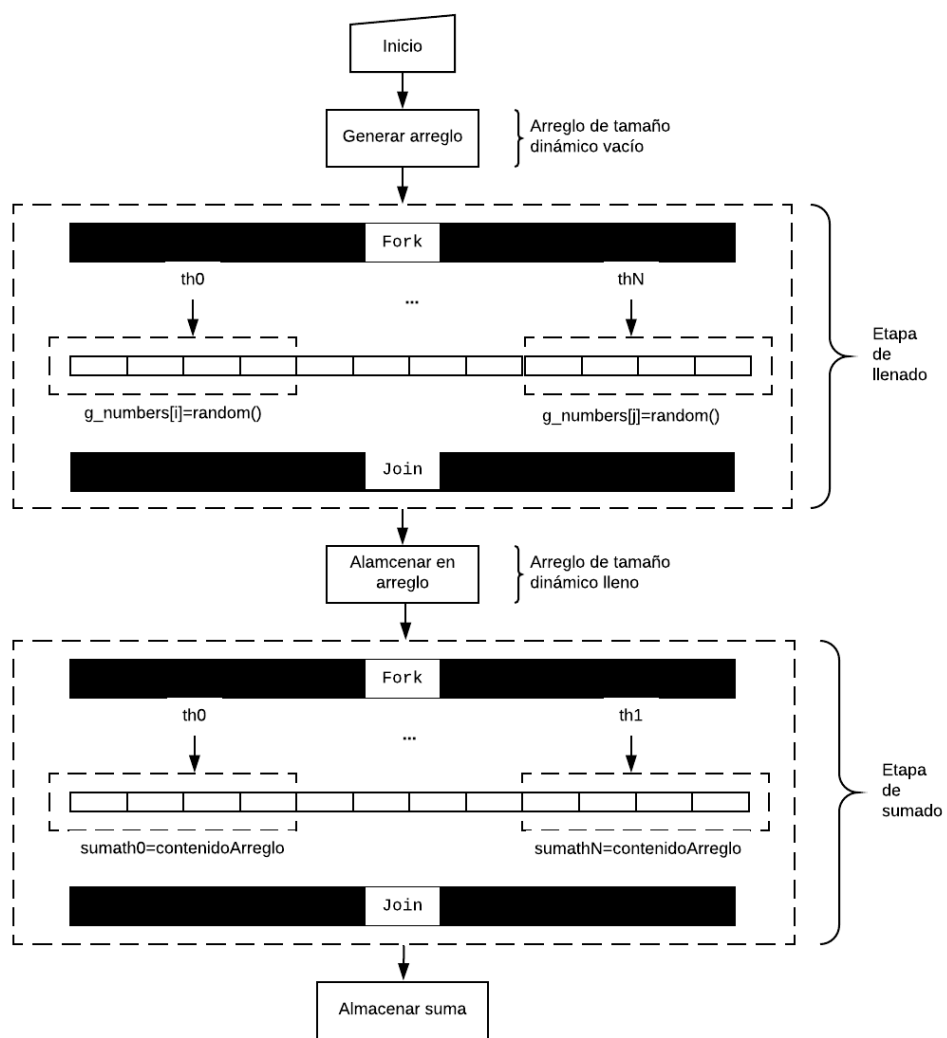


Figura 1 Diagrama general de solución.