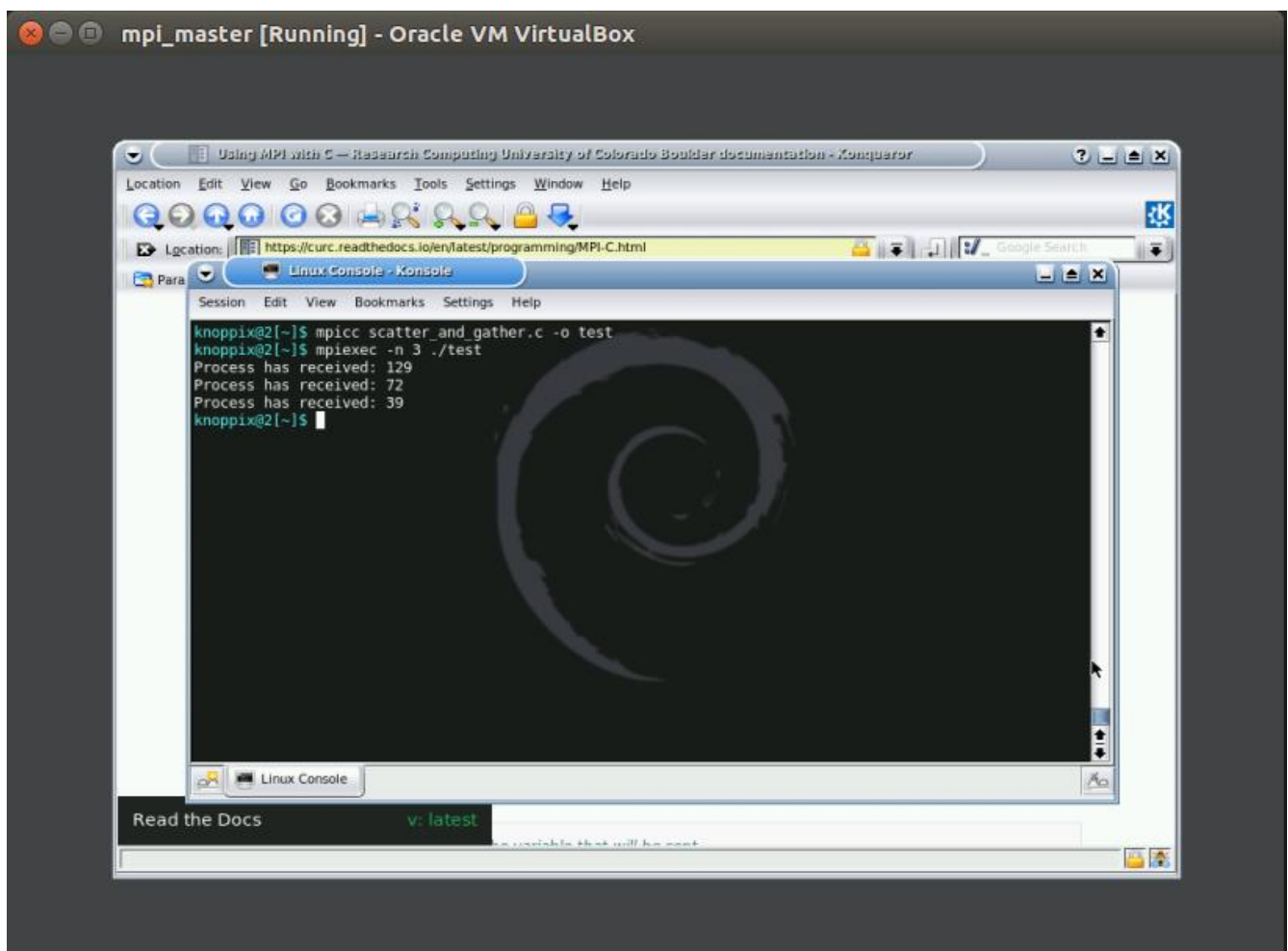


Prueba dos ejemplos de OpenMPI

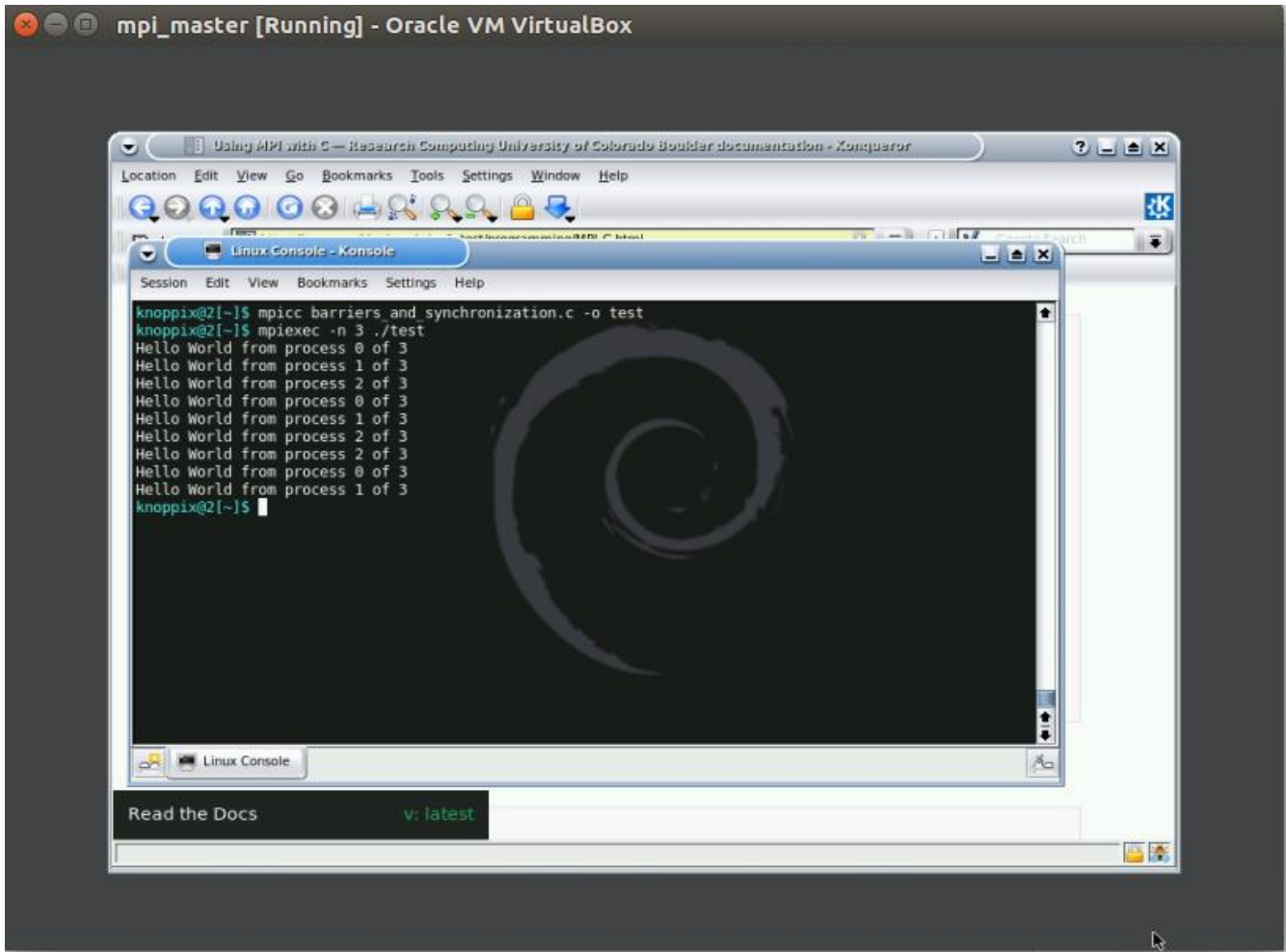


MPI. Permiten que las franjas de datos se distribuyan desde un proceso raíz a todos los demás procesos disponibles, o los datos de todos los procesos se pueden recopilar en un solo proceso. Estos operadores pueden eliminar la necesidad de una cantidad sorprendente de código reutilizable mediante el uso de dos funciones

Con el fin de obtener una mejor comprensión de estas funciones, vamos a seguir adelante y crear un programa que utilizará la función de dispersión. Tenga en cuenta que la función de recopilación (no se muestra en el ejemplo) funciona de forma similar y es esencialmente la inversa de la función de dispersión. Otros ejemplos que utilizan la función de recopilación se pueden encontrar en los tutoriales de MPI enumerados como recursos al principio de este documento.

Crearemos un programa que dispersa un elemento de una matriz de datos en cada proceso. Específicamente, este código dispersará los cuatro elementos de una matriz en cuatro procesos diferentes. Comenzaremos con una función principal básica de C++ junto con variables para almacenar el rango del proceso y el número de procesos.

Al ejecutar este código(`scatter_and_gather.c`) imprimirá los cuatro números de la matriz de distribución como cuatro números separados cada uno de diferentes procesadores (tenga en cuenta que el orden de los rangos no es necesariamente secuencial)



La sincronización es una herramienta esencial en la seguridad de subprocesos y garantizar que ciertas secciones de código se manejen en ciertos puntos. es un bloqueo de proceso que mantiene cada proceso en una determinada línea de código hasta que todos los procesos han alcanzado esa línea en el código.

En este código podremos ver como nuestro programa "Hello World" imprimira cada proceso en orden de identificación de subprocesos. Comenzando con nuestro código "Hello World" de la sección anterior, comience anidando nuestra instrucción print en un bucle.

La función de barrera en el bucle. Esto garantizará que todos los procesos se sincronicen al pasar a través del bucle.