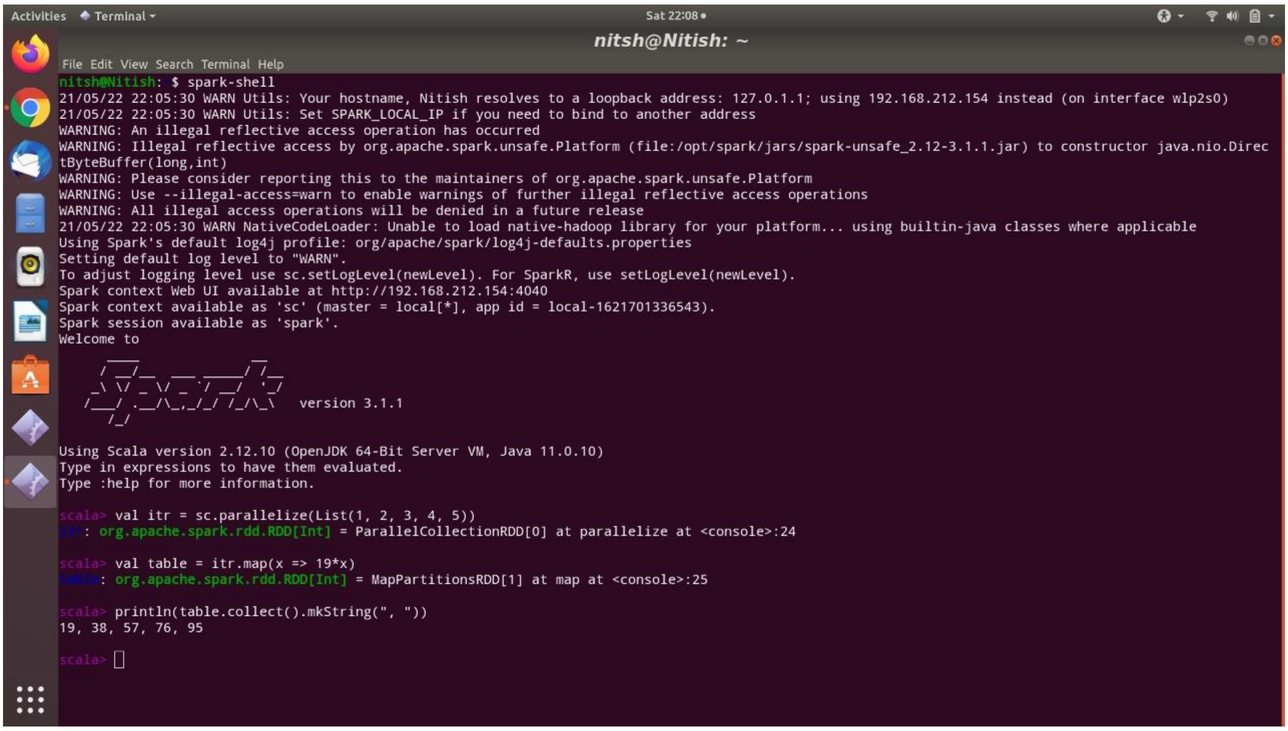
BDA Assignment 4

- **map()**



```
nitsh@Nitish: $ spark-shell
21/05/22 22:05:30 WARN Utils: Your hostname, Nitish resolves to a loopback address: 127.0.1.1; using 192.168.212.154 instead (on interface wlp2s0)
21/05/22 22:05:30 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/opt/spark/jars/spark-unsafe_2.12-3.1.1.jar) to constructor java.nio.Direc
tByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
21/05/22 22:05:30 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://192.168.212.154:4040
Spark context available as 'sc' (master = local[*], app id = local-1621701336543).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.1
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 11.0.10)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val itr = sc.parallelize(List(1, 2, 3, 4, 5))
itr: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val table = itr.map(x => 19*x)
table: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[1] at map at <console>:25

scala> println(table.collect().mkString(", "))
19, 38, 57, 76, 95

scala> 
```

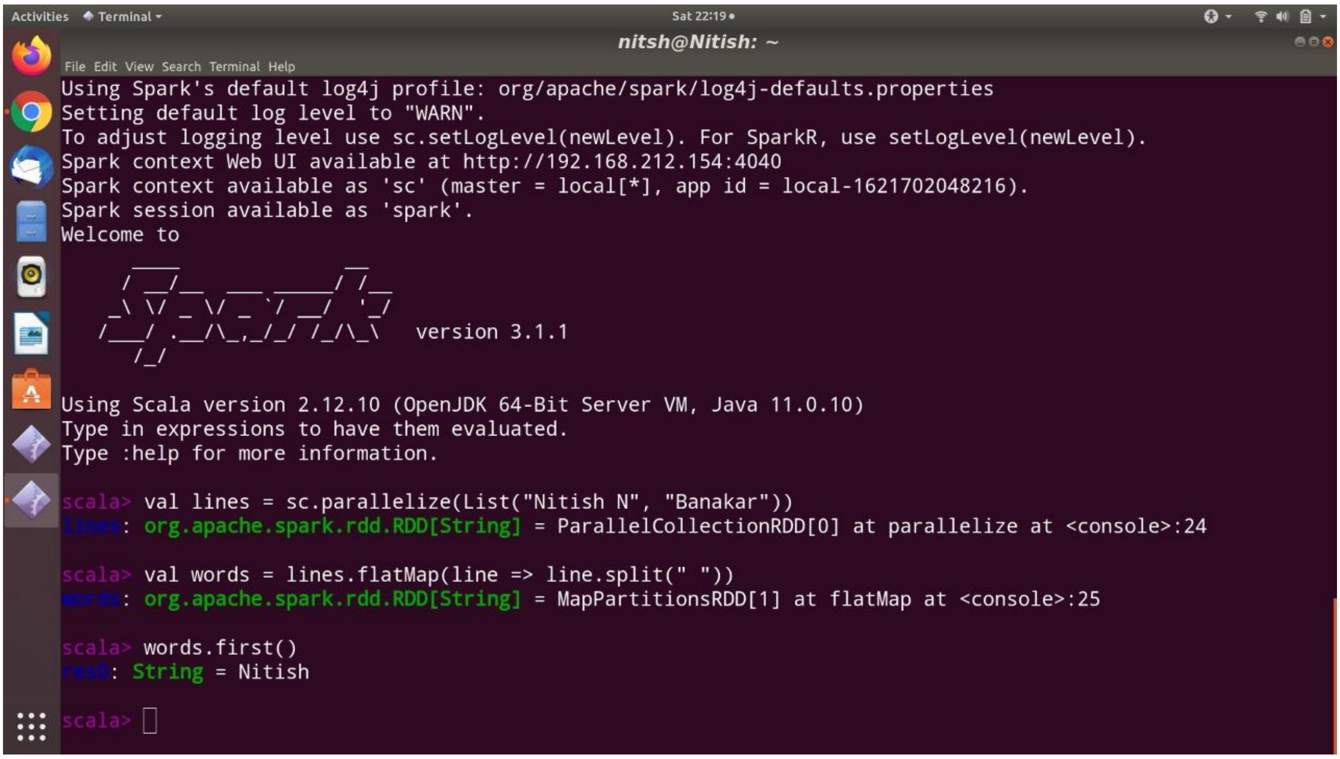
code:

```
scala> val itr = sc.parallelize(List(1, 2, 3, 4, 5))
itr: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val table = itr.map(x => 19*x)
table: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[1] at map at <console>:25

scala> println(table.collect().mkString(", "))
19, 38, 57, 76, 95
```

- **flatmap()**



```
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://192.168.212.154:4040
Spark context available as 'sc' (master = local[*], app id = local-1621702048216).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.1
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 11.0.10)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val lines = sc.parallelize(List("Nitish N", "Banakar"))
lines: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val words = lines.flatMap(line => line.split(" "))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[1] at flatMap at <console>:25

scala> words.first()
res0: String = Nitish

scala> 
```

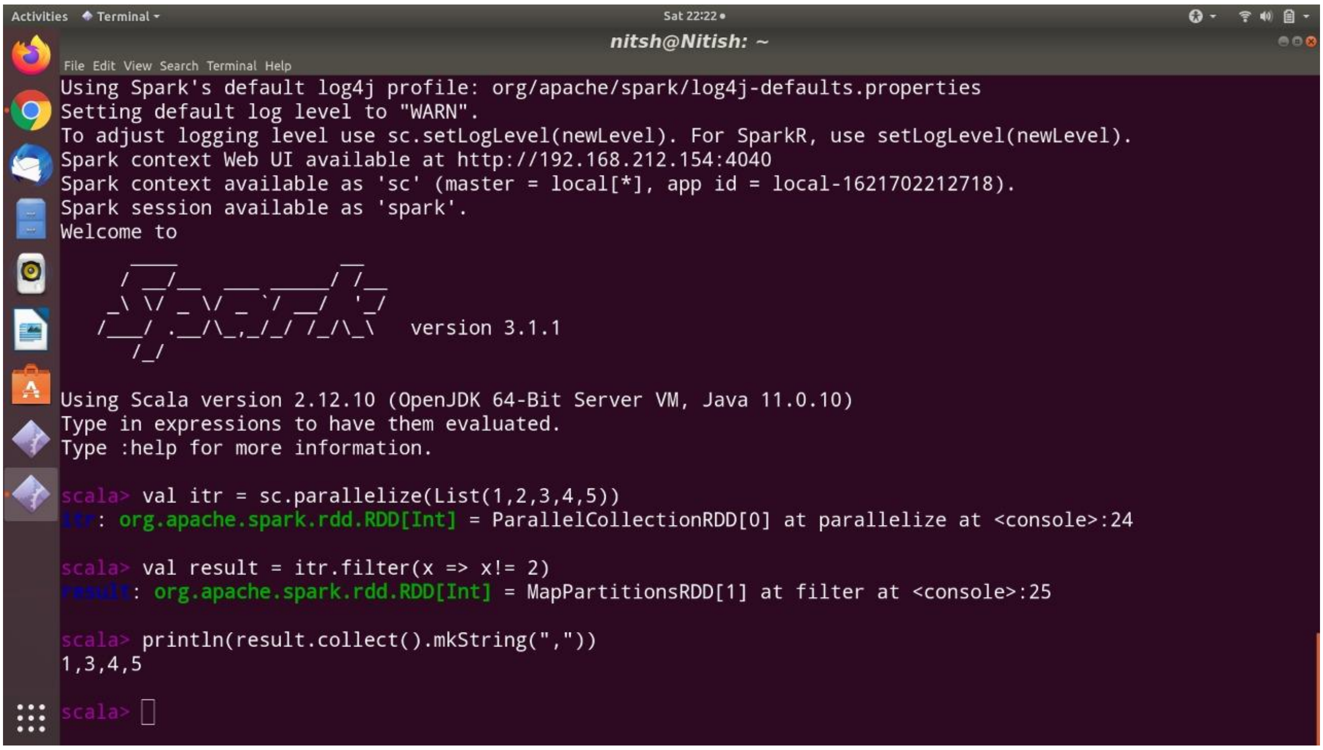
code:

```
scala> val lines = sc.parallelize(List("Nitish N", "Banakar"))
lines: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val words = lines.flatMap(line => line.split(" "))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[1] at flatMap at <console>:25

scala> words.first()
res0: String = Nitish
```

- **filter()**



```
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://192.168.212.154:4040
Spark context available as 'sc' (master = local[*], app id = local-1621702212718).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.1
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 11.0.10)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val itr = sc.parallelize(List(1,2,3,4,5))
itr: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val result = itr.filter(x => x!= 2)
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[1] at filter at <console>:25

scala> println(result.collect().mkString(","))
1,3,4,5

scala> 
```

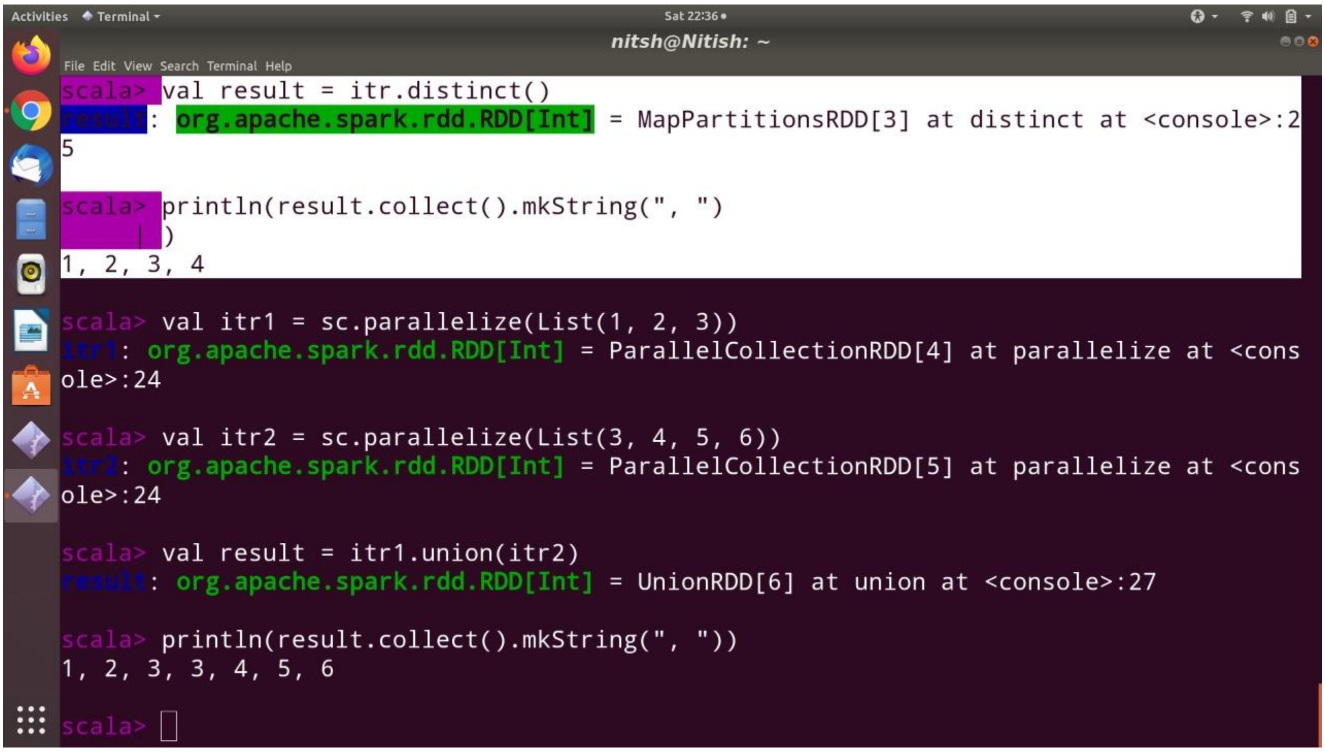
code:

```
scala> val itr = sc.parallelize(List(1,2,3,4,5))
itr: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val result = itr.filter(x => x!= 2)
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[1] at filter at <console>:25

scala> println(result.collect().mkString(","))
1,3,4,5
```

- **union()**



```
scala> val result = itr.distinct()
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[3] at distinct at <console>:2
5

scala> println(result.collect().mkString(", ")
     | )
1, 2, 3, 4

scala> val itr1 = sc.parallelize(List(1, 2, 3))
itr1: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[4] at parallelize at <cons
ole>:24

scala> val itr2 = sc.parallelize(List(3, 4, 5, 6))
itr2: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[5] at parallelize at <cons
ole>:24

scala> val result = itr1.union(itr2)
result: org.apache.spark.rdd.RDD[Int] = UnionRDD[6] at union at <console>:27

scala> println(result.collect().mkString(", "))
1, 2, 3, 3, 4, 5, 6

scala> 
```

code:

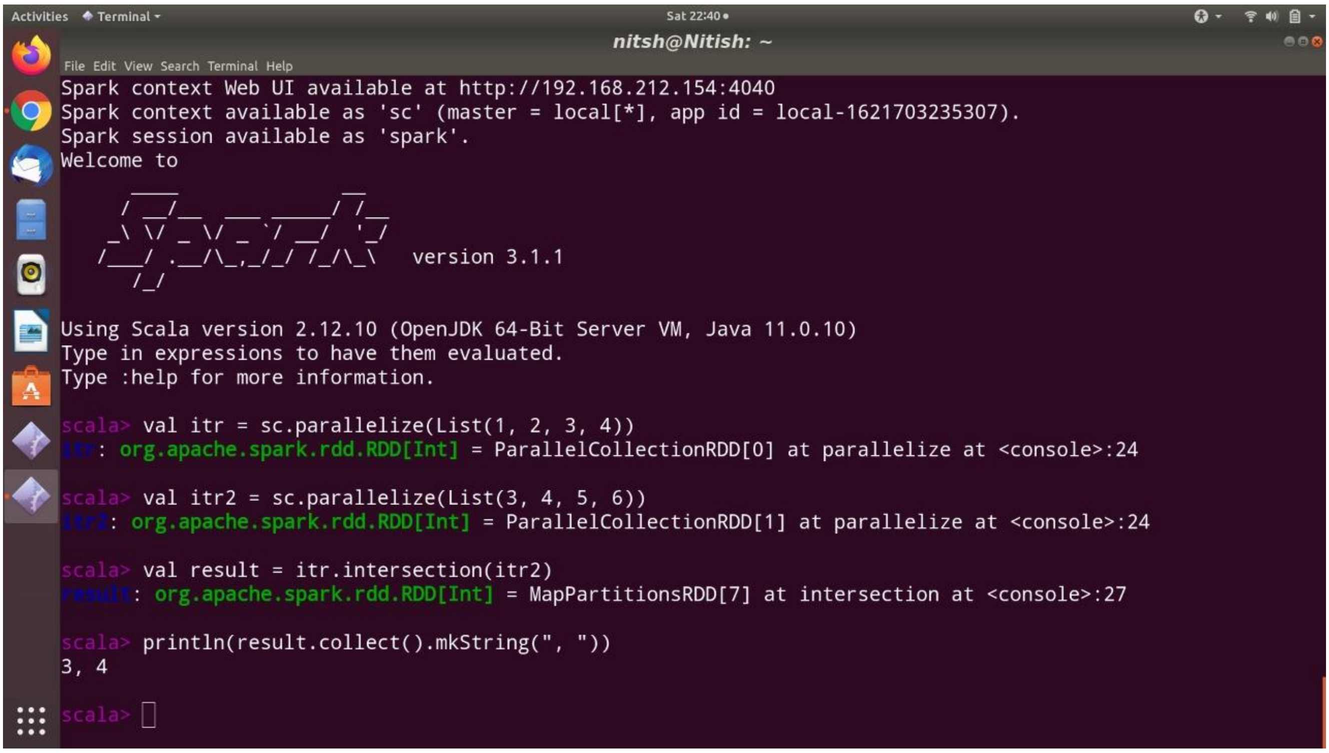
```
scala> val itr1 = sc.parallelize(List(1, 2, 3))
itr1: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[4] at parallelize
at <console>:24

scala> val itr2 = sc.parallelize(List(3, 4, 5, 6))
itr2: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[5] at parallelize
at <console>:24

scala> val result = itr1.union(itr2)
result: org.apache.spark.rdd.RDD[Int] = UnionRDD[6] at union at
<console>:27

scala> println(result.collect().mkString(", "))
1, 2, 3, 3, 4, 5, 6
```

- **intersection()**



```
Spark context Web UI available at http://192.168.212.154:4040
Spark context available as 'sc' (master = local[*], app id = local-1621703235307).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.1
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 11.0.10)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val itr = sc.parallelize(List(1, 2, 3, 4))
itr: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val itr2 = sc.parallelize(List(3, 4, 5, 6))
itr2: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[1] at parallelize at <console>:24

scala> val result = itr.intersection(itr2)
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[7] at intersection at <console>:27

scala> println(result.collect().mkString(", "))
3, 4

scala> 
```

code:

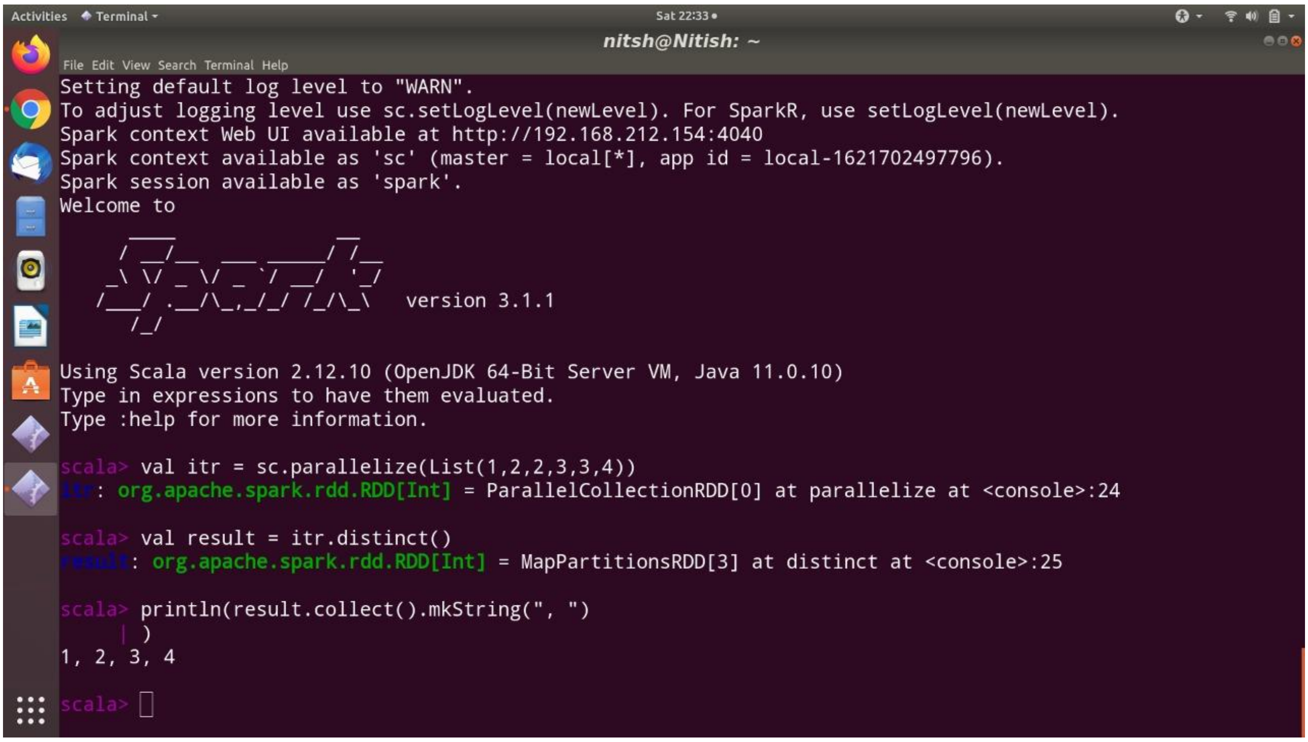
```
scala> val itr = sc.parallelize(List(1, 2, 3, 4))
itr: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val itr2 = sc.parallelize(List(3, 4, 5, 6))
itr2: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[1] at parallelize at <console>:24

scala> val result = itr.intersection(itr2)
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[7] at intersection at <console>:27

scala> println(result.collect().mkString(", "))
```

- **distinct()**



```
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://192.168.212.154:4040
Spark context available as 'sc' (master = local[*], app id = local-1621702497796).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.1
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 11.0.10)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val itr = sc.parallelize(List(1,2,2,3,3,4))
itr: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val result = itr.distinct()
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[3] at distinct at <console>:25

scala> println(result.collect().mkString(", ")
     | )
1, 2, 3, 4

scala>
```

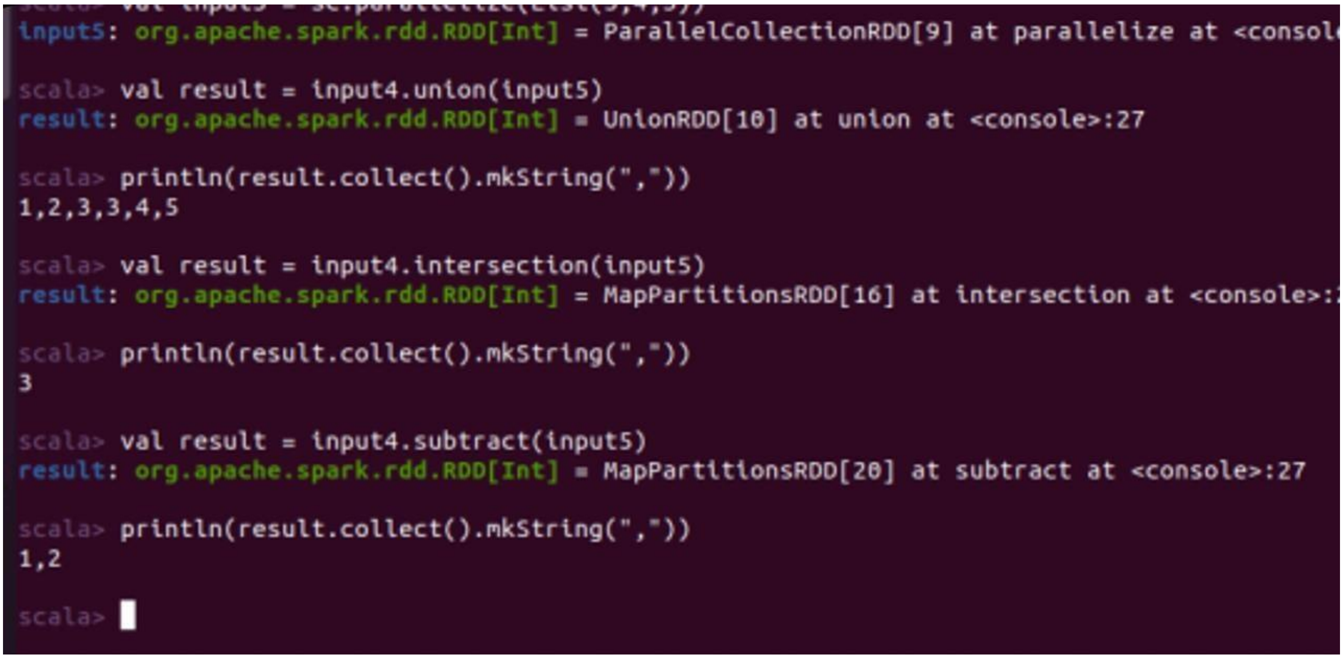
code:

scala> val itr = sc.parallelize(List(1,2,2,3,3,4))
itr: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val result = itr.distinct()
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[3] at distinct at <console>:25

scala> println(result.collect().mkString(", "))
1, 2, 3, 4

- **subtract()**



```
scala> val input5 = sc.parallelize(List(3,4,5))
input5: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[9] at parallelize at <consol

scala> val result = input4.union(input5)
result: org.apache.spark.rdd.RDD[Int] = UnionRDD[10] at union at <console>:27

scala> println(result.collect().mkString(","))
1,2,3,3,4,5

scala> val result = input4.intersection(input5)
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[16] at intersection at <console>:

scala> println(result.collect().mkString(","))
3

scala> val result = input4.subtract(input5)
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[20] at subtract at <console>:27

scala> println(result.collect().mkString(","))
1,2

scala> █
```

<u>code:</u>

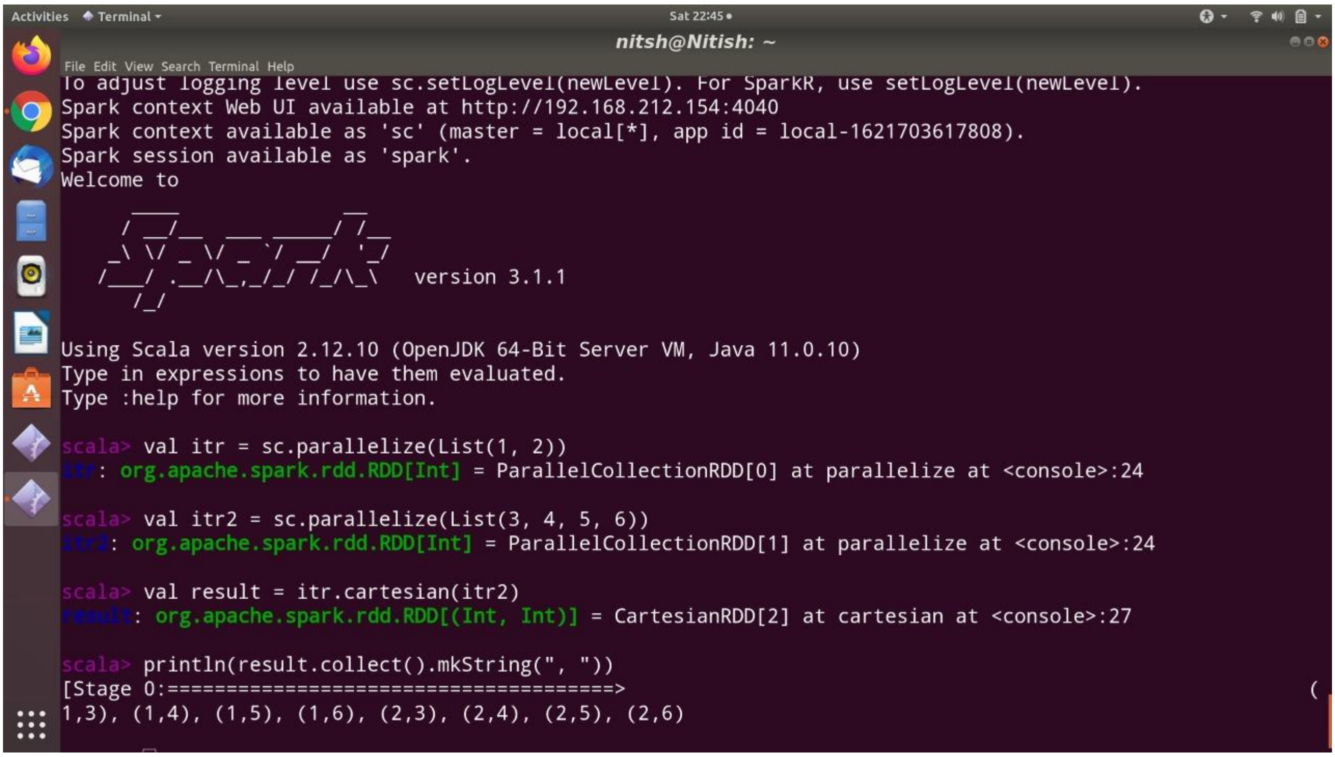
```
scala> val itr1 = sc.parallelize(List(1, 2, 3, 4))
itr1: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize
at <console>:24

scala> val itr2 = sc.parallelize(List(3, 4, 5, 6))
itr2: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[1] at parallelize
at <console>:24

scala> val result = itr1.subtract(itr2)
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[5] at subtract at
<console>:27

scala> println(result.collect().mkString(", "))
1, 2
```

- **cartesian()**



```
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://192.168.212.154:4040
Spark context available as 'sc' (master = local[*], app id = local-1621703617808).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.1
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 11.0.10)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val itr = sc.parallelize(List(1, 2))
itr: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val itr2 = sc.parallelize(List(3, 4, 5, 6))
itr2: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[1] at parallelize at <console>:24

scala> val result = itr.cartesian(itr2)
result: org.apache.spark.rdd.RDD[(Int, Int)] = CartesianRDD[2] at cartesian at <console>:27

scala> println(result.collect().mkString(", "))
[Stage 0:=======================================>                                        (
1,3), (1,4), (1,5), (1,6), (2,3), (2,4), (2,5), (2,6)
```

<u>code:</u>

```
scala> val itr = sc.parallelize(List(1, 2))
itr: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize
at <console>:24

scala> val itr2 = sc.parallelize(List(3, 4, 5, 6))
itr2: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[1] at parallelize
at <console>:24

scala> val result = itr.cartesian(itr2)
result: org.apache.spark.rdd.RDD[(Int, Int)] = CartesianRDD[2] at cartesian
at <console>:27

scala> println(result.collect().mkString(", "))
[Stage 0:=====================================>
(1,3), (1,4), (1,5), (1,6), (2,3), (2,4), (2,5), (2,6)
```