**Project Members :**

Jaimin Bhatt [3819-8457]

Srinivas Balaji [9421-7395]

**Files:**

***actorlog.scala -***

The object "actorlog" has to be imported in the main scala file in order to use the logging facility.

It contains the following functions for creating and writing the log files:

write\_Log

start\_Log

stop\_Log

send\_Log

receive\_Log

generateFilename

These functions perform different logging operations depending on the parameters passed.

***example.scala -***

The example program demonstrates simple message passing between actors. A boss actor creates a number of worker actors (defined 10 in the code) and also generates a number of different messages (defined 10 in the code). Boss then randomly selects a worker and sends it the message. Each worker forward the message randomly to another worker. The message finally gets delivered when it has passed to all the available actors.

Predefined number of actors =10 & messages = 10.

It uses the functions present in the "actorlog.scala" for creating separate log files for each of the actors and messages being passed.

**How to compile:**

The files can be executed using the following commands:

$ scalac actorlog.scala example.scala

$ scala example

**Log files:**

In "actorlog.scala" we have provided the functions to creating a log for each actor including the boss and also the option to create a log for each of the messages for tracing the flow of the message between the actors.

A program can use either of the two logging options depending on its functionality.

In case of "example.scala" we use both the actor and message logging facility.

Hence executing "example.scala" will result in creation of separate log files for each actor created and message being passed.

Example of the log files created:

Actor\_1\_Log.txt -

11:20:2012:09:58:50:948---Actor1 Started

11:20:2012:09:58:50:963---Message2 Received By : Actor1

11:20:2012:09:58:50:963---Message2 Sent By : Actor1 to : Actor4

11:20:2012:09:58:50:964---Message3 Received By : Actor1

11:20:2012:09:58:50:964---Message3 Sent By : Actor1 to : Actor9

11:20:2012:09:58:50:965---Message1 Received By : Actor1

11:20:2012:09:58:50:965---Message1 Sent By : Actor1 to : Actor9

11:20:2012:09:58:50:965---Message8 Received By : Actor1

11:20:2012:09:58:50:965---Message8 Sent By : Actor1 to : Actor8

11:20:2012:09:58:50:965---Message5 Received By : Actor1

11:20:2012:09:58:50:966---Message5 Sent By : Actor1 to : Actor9

11:20:2012:09:58:50:966---Message4 Received By : Actor1

cont....

Msg\_3\_Log.txt -

11:20:2012:09:58:50:957---Message3 Created

11:20:2012:09:58:50:958---Message3 Sent By : Boss Actor to : Actor4

11:20:2012:09:58:50:959---Message3 Received By : Actor4

11:20:2012:09:58:50:959---Message3 Sent By : Actor4 to : Actor1

11:20:2012:09:58:50:964---Message3 Received By : Actor1

11:20:2012:09:58:50:964---Message3 Sent By : Actor1 to : Actor9

11:20:2012:09:58:50:965---Message3 Received By : Actor9

cont....

**BONUS :**

***actorlog\_bonus.scala -***

The object "actorlog\_bonus" contains a trait which contains the following functions for creating and writing the log files:

It contains all the functions that were present in “actorlog” only the following additional function is added:

create\_CombinedLog

***example\_bonus.scala –***

Is similar to “example.scala”

**How to compile:**

The files can be executed using the following commands:

scalac actorlog\_bonus.scala example\_bonus.scala  
scala example\_bonus

**Log files:**

All the log file that are created in the case of “example.scala” are also created for “example\_bonus.scala”.

On top of that a file “CombinedLog.txt” is created that contains the chronologically combined log of all the actors.