**Lab 1: Portable, Migratable Work**

**Abhishek Bhowmik (abhowmi1), Neil Rajesh Dhruva (ndhruva)**

**Contents**

**Introduction …………………………………………………………………………….**

**Problem Definition …………………………………………………………………..**

**Solution Overview ……………………………………………………………………**

**Framework ………………………………………………………………………………**

**Code Organization ……………………………………………………………………**

**Conclusion ……………………………………………………………………………….**

**Introduction**

**Problem Definition**

The problem deals with building a distributed system that can migrate processes from one node to another, while preserving their state. The deliverables include:

1. Creating a framework for migratable processes.
2. This project report, which outlines the framework by providing information about the design and implementation of the system.
3. Test and example code for testing the system.

The distributed system must be able to preserve the state of file input/output operations, as well as the local state of the processes running on each of the nodes. It must also keep track of the various nodes that are part of the system.

Additionally, a user interface has to be provided which gives the user the freedom to:

1. Create a new process with a desired set of arguments on any of the active nodes.
2. Migrate a process form one node to another. The process has to restart from where it left off.
3. Get a list of processes and the information about the nodes they are running on.

**Solution Overview**

For creating a distributed system that preserves process state while migrating it from one node to another, we have created a model similar to the ‘Master/slave’ model. The master interacts with the user and gets the commands. It then informs a slave to perform those commands as per user request.

The interaction between the master and user is limited to taking the command as an input, and printing reports of the current state of processes and slaves (nodes) to the user.