CISC 3320 Homework 2 Design Document

Michael Rocco

Module 1:

public static int allocate\_map()

Uses protected static class Bits with a BitSet et for storing PIDs. Class is inaccessible outside Cis3320hw2.

It sets the value one higher than constant MAX\_PID as true and lets the BitSet class handle pre-allocation of space for PIDs.

It returns 1 if successful, -1 if it fails.

Module 2:

public static synchronized int allocate\_pid()

return new process ID from BitSet et in Bits class. Returns first available.

Returns -1 if there are no available PIDs. Synchronized because BitSets are not thread safe.

Module 3:

public static synchronized void release\_pid(int pid, BitSet pa)

Releases PID at supplied integer for use by another process. Check if PID is in use, if it is Sets boolean value at BitSet position pid as false. Prints an error to console if an attempt to free an unused PID is made. Synchronized because BitSets are not thread safe.

Module 4:

protected static class Bits{}

contains static BitSet et that stores PIDs for program.

Module 5:

private static class RunnableImpl implements Runnable{}

Private class of Cis3320hw2, contains internal int pid1. Used to make multiple threads instead of doing it in main.

pid1 is used to allocate a new pid using allocate\_pid. The thread waits a random time up to 100 milliseconds. Pid1 is then freed using release\_pid.