Operating Systems (Extended)
Semester 1, 2014
Eike Ritter

The University of Birmingham
12 December, 2014

Assessed Exercise sheet 1

Deadline: January 16, 12 noon

For this exercise we consider a bespoke operating system for a robot. We consider a robot with several arms which is in addition mobile. Each arm is controlled by a separate motor. In addition there are sensors which are used to calculate the position of the robot. The choice of movements is done by a planning process, which analyses the current position and movement of the arms and initiates further movements.

To avoid damage, the robot has an emergency stop function, which stops the robot and halts the movements of all arms if certain conditions are met (eg the robot comes too close to a wall). A very short response time is essential for this stop function.

The robot also contains flash memory as stable storage, normal memory but not MMU (so no hardware support for memory management), and mechanisms for handling interrupts, together with sensors and motors.

The robot needs to execute the following tasks:

- reading the sensor data
- stopping the robot
- moving a robot arm or the whole robot
- planning further movements

Also assume that there is not sufficient main memory to hold the data required for the planning process in main memory. However, there is sufficient space available in flash memory.

1. Describe which processes and which threads you would create to execute all the given tasks;
2. Describe in detail suitable scheduling strategies for this scenario. If you use priorities, you should indicate precisely how you assign priorities.
3. Describe in detail a suitable memory management strategy for this scenario. You should in particular describe strategies for allocating memory to processes and/or threads, and also criteria for swapping. Your solution should maximise the degree of multiprogramming. You need to take into account that the response time for the task which stops the robot is smaller than the time required to swap. Also, once the motors are started, they require a response time which is also smaller than the time required to swap.

Note in particular the dependencies between the answers given to the different parts. You should make sure that interactions between the division into processes and threads, scheduling and memory management strategies are taken into account.