CSEII - Project Report
Group 11

In this report we will explain which tests we've used in this project and what exactly they test. There are two tests (which the course provides), one which tests the functions terminate(), run(), init_kernel() and create_task(), and the other ones tests the communication functions such as recieve_wait(), send_no_wait() etc.

Testfile 1 - terminate(), run(), init_kernel() & create_task():
As stated above, here the file tests if it can terminate a running task and if the kernel can create a task without causing any forms of errors or memory leaks etc. This file mainly focuses on the create_task() function as it's one of the head functions which makes the kernel work. Firstly the testfile checks if init_kernel() works which basically starts the kernel. Once that's out of the way the file checks for dummy nodes in the lists ReadyList, WaitingList and TimerList. A certain amount of tasks is being created with a low deadline to check how the kernel handles it if it fails or not to do so. If even a single task fails to create, the whole program will fail. At the end of the testfile the file tries to create a task when called during run time. task_body_2 will create tasks with tighter deadlines than the currently running task. The file creates a task with a higher deadline (while the ones created right now have a low deadline) and if it's successfully created it goes to the last stage of the testfile and tries to create a task withe a very high deadline which has higher priority than the idle task but lower than all the other tasks (task_body_3 is used here and is the end station of the test if create_task() can use the task body as a task body). If this is succeeded the testfile is complete and these four functions work.

Testfile 2 - Communication functions
This file tests all the communication functions which basically takes care of the tasks to send and receive them and check if the mailboxes are working. Firstly it checks if the kernel is initialized or not and also check for dummy nodes. The test creates three mailboxes which has a type of its own, a integer, a char and a float and checks if the mailbox is created (if it doesn't return NULL), if the mailbox doesn't create it will fail. The file will then as an end of the test try to create three tasks with different bodies (which tests different functions), task_body_1, task_body_2 and task_body_3 (note that if anything explained in these bodies goes in the opposite direction the test will fail). The task_body_1 task is created with a low deadline and the body tests on an empty mailbox how recieve_no_wait works three times to check if it works properly. It later goes and tries to send to the mailbox with send_no_wait and check if it actually works, after that it's the same but with recieve_no_wait and checks if it receives and reads properly. At last the test tries to send_wait to the integer mailbox, if it doesn't fail it will go to the second send_wait which tries to do it with the float mailbox, and here it is supposed to fail otherwise we fall into a infinite while-loop. Now task_body_1 is complete (terminate() used at the end). In task_body_3 a Data of 500 is being sent to the integer mailbox with send_wait and then terminate(). In task_body_2 the test is with recieve_wait, it will check with the integer mailbox that the Data sent in in these two scenarios isn't 100 (varInt_t2) or 500 (retVal_t2), if they are not any of these numbers the test file has been passed.