# *Assignment 1: Processes and Threads*

# Table of Contents:

|  |  |
| --- | --- |
| Page | Number |
| Statement of Completeness | 1 |
| Description of Data Structures and High Level Design | 1 |
| Output Listing | 2 - 5 |

# Statement of completeness:

I attempted all tasks besides synchronization defined in the specification . I tried to accurately model the output screenshots given, however my date and time is slightly different to yours. I was also unable to emulate the keyboard input without requiring a return character and an echo. The timestamp duration (and several bits of functionality related) is also incomplete.

The arrival queue works perfectly (as far as I know), however I struggled with the enter carpark and depart carpark threads. I opted NOT to use a linked list, and regretted this as I realised random removal of cars was required. My departure function does not correctly remove cars, instead marking them null.

# Descriptions of data structures and high level design:

I chose to implement a queue data structure for the arrival queue. It contains several basic functions (EnQueue, DeQueue, empty, full etc). It’s defined in the queue.c file, and works perfectly. We were allowed to use either a fixed buffer array or a linked list to emulate the carpark. I decided to use a fixed array think it would be more straightforward, but regretted my decision as I realised we had to remove the cars randomly. I could not figure out an elegant way to allow this. I chose to fill the carpark array with NULL values on startup. Every time a car was to be added, it first iterated through all the parks in the carpark looking for a NULL valued one. It then parked itself there. On removal, I simply reset the car to NULL (thus marking the carpark park as empty). I thought this to be a fairly clumsy solution, and wished I’d gone with a linked list which would have made things enormously more straightforward.

I also saw several functions defined in the skeleton template given, and it was not immediately obvious what purpose they were intending on performing. I opted the leave them out if I couldn’t see their purpose. I did not see there was a remove\_car function prototype, and unfortunately left it out.