Rubric Assignment 1 Operating Systems

	Insufficient	Mediocre	Sufficient	Excellent
Correctness of the output	Input gives incorrect output.	Input gives the correct output, but in the wrong order.	Input gives correct output.	Input gives correct output, and the code is designed in such a way that is can be extended in a logical way and corner cases are handled or explicitly noted.
Documentation of the code	No documentation.	Unclear documentation.	Small hints on essential places.	Clear hints that clarify the structure of the processes.
Structure of the code	Unclear naming of variables, no clear subdivision into functions.	Clear naming scheme, but unclear subdivision into functions (e.g. long functions)	Clear naming scheme and structure.	Clear naming scheme and structure which is easy to extend.
Mastery of system calls (documentation of system calls)	System calls are used in an unstructured (trial and error) way.	System calls are used, but not in the right order.	The right system calls are used, but the corresponding administration/memory maintenance aren't done properly.	The right system calls are used, and the corresponding administration/memory maintenance are correct.
The way processes are controlled	Too many unnecessary processes are created.	The created processes have an unclear structure.	The created processes have a clear structure.	The created processes have a clear structure, which is optimal.
Reflection	The reflection doesn't show insights.	The reflection shows some insights, but it doesn't show a clear overview of different ways in which the product could have been implemented.	The reflection shows some insights, and it shows a clear overview of different ways in which the product could have been implemented. Reliability/testing is not discussed.	The reflection shows insights in a sufficient way, and it shows a clear overview of different ways in which the product could have been implemented. Disadvantages of the used approach and possible alternatives are discussed, as well as reliability/testing.