Problem Description

Group member:

Alastor Liu; 17al88; 20112170

Xiangman Li; 18xl4; 20119884

Firstly, we created a function init_module to determine if the element is readable, if it is Null, return -1, else we set the proc_entry into my_read_proc. The cleanup_module function is to set the entry into Null so that nothing is in there.

Secondly, in my_read_proc, we used 'if else' to check if the process starts at the beginning of the file (when fpos equals 0). Path 1, if the process starts at the beginning, we print "Hello World" as well as the column titles "PID UID VSZ RSS" using numChars by adding each string into it. In the first path we used a while loop to check the priority (PID) of each task, If PID is 0, we move on to the next task. We then used another 'if else' to check the status of the memory information (mm), if mm is null, we add two 0s to the buffer, otherwise, we multiply total_vm and rss contained in the memory information by the page size and print them.

Lastly, in path 2 (under big else), when the pointer is back to the beginning, we use the same method as before to print the information in memory. We set *eof to 1 and *start to page and return numChars at the end of the function.

The one special trick that we used is that we used a circular linked list, which has its last element pointing to its first element.