CSCI 451 – Assignment 6

Grant Haataja

Case 1:

Upon encountering the & symbol, the program called pthread\_exit(NULL) and thread 2 exited. The program locked up after that, neither exiting nor continuing to process any data. The reason for this is because after exiting thread 2, Main never received a signal to continue writing characters to the output file, and thus became in a state of indefinite postponement, waiting forever.

Case 2:

Upon encountering the & symbol, the program called “kill -s 9 TID”. I actually did this in two different ways. When I used pthread\_self() it returned an unsigned long integer, so the “kill” command did not execute properly, giving me an illegal number error. The program continued processing the data and ended in a state of waiting at the end since the number of characters in each file did not match up. When I switched the logic to use the Linux system call “gettid” to obtain the TID of Main, using that in the “kill” command successfully stopped the program from running upon entering the & symbol.