Pawel Derkacz – pd304
Deep Patel – dp767

ReadMe for Assignment 2

Concerning compressT_LOLS.c:

Our threading algorithm relies on the number of bytes the file contains in **total,** in order to assign how many bytes each thread should read from that file. Each byte may not be an alphabetical number, therefore the compression may **look** unevenly distributed even though each thread received an order to read the same amount (except for the first, which takes on any excess bytes via modulus calculation). A struct is used in order to pass in arguments for each thread. The struct contains that thread's ID, thread number (number from the order of spawned threads), the starting location from where the thread should read, the length of bytes it should read, the original filename passed from argv, and the partially modified filename to where the thread should output.

As per Dr. Tjang's request, main is only used to initiate the program launcher "launch" which takes in the same parameters as main would, but returns nothing.

Concerning compressR_worker_LOLS.c:

The main of this program is very nearly the same as the encoder portion of the threading program. Here are the changes:
1) The struct no longer exists, instead everything is passed as a char * through argv.
2) A few names were changed around to reflect that the program uses multi-processing and not threading.

In general, the multiprocessing program follows the same exact procedures as explained above. There may be an issue where passing in 1 for number of pieces to split to file into (meaning, not splitting at all), still causes the output file to have a 0 tacked onto it. No idea as to why, since there is an if statement checking for just this scenario.

Concerning compressR_LOLS.c:

Similarly to the threading program, this program uses a function "launch" as a launcher to process splitting and waiting. This parent program follows nearly identical initialization procedures that the threading program used, with added conversions from int to char *, so that the arguments can be passed onto the child.

TL;DR

Both main/parent files **use** a function "**launch**" in order **to** help **simplify grading,** **if** grading from a **personal piping program** that **doesn't use main**. Length to **read** document is **by bytes**, **not** necessarily **characters**: **output may look different** from other persons**, but** should **still** be **correct.**