

Compress Assignment

For our compressed program we have 3 files which will respectively be using threads, parent process and child process. In the thread file the compress function first checks for any errors. It then opens the file and checks the size of the file. After that we determine the size of each partition and the size of the first partition as well (since the first partition will get the remainder). We also create a struct to store the parameters that each thread will use, since `pthread_create` only accepts one argument. We then create an array of tids that will store each instance of `pthread_create`. From here, threads will take over and call the LOLS function. LOLS simply creates a new filename, and then compresses the section of code given to it through the struct arguments. This is done easily by using a for loop which starts at the given start position and ends at the start position plus the partsize. Because we pass arguments through a struct which is stored in the heap, the arguments are subject to changing before each respective thread can store its values. This led to us using a condition variable that waited for a signal from the LOLS function. The signal is sent after all arguments are stored in the local variables of the function. After this, the threads are all joined and threading is completed. For our processes we had to create 2 c programs since one of them is the parent while the other is the child that does most of the work. In our child C program, we create new compressed files and then right after that we call upon the RLE algorithm that checks each character to see if it is an alphabet and then keeps count of it. Once it reaches a new letter then it will reset it and sent what the old counter was to the desired compressed file. The parent file basically calculates what the offset is and takes in the uncompressed file and then sends that info to the child. It sends that information through the `exec()` function since calls upon a different executable and does what that executable demands.