Elikem Hermon | 20075527
Hermann Krohn | 20057435

Lab 1 – Description of Program

The program starts in the main function by reading the /proc directory using the scandir function. This function accepts four arguments, namely, a directory, a pointer that holds the listing of subdirectories, a function (isProcessDir) that filters subdirectories, and a sorting function which is NULL in this case. The scandir function returns the number of process directories that were read. If it returns -1, there is an error and the program terminates.

The isProcessDir function determines if a directory is a process by checking if the folder name is numeric. This is implemented by first making a pointer to point to the directory name. The pointer iterates through every character of the name and calls the isdigit function to determine if the current character is numeric. If every character is numeric, the isProcessDir function returns true, otherwise false.

Next, the headings for the output are printed. Then the program prints pid, name, status, user, and group for each process. To do this, the program iterates through each process and generates a path to its status file using the sprintf function. The file is then opened using fopen with the read option. If fopen returns null, the program terminates as there is an error.

The program then iterates through all lines in the status file and selects the desired data (name, status, user, group). To achieve this, the program reads each line into a buffer using fgets and checks if it contains any of the desired data. If the desired data is found, the new line character is removed, and the data is copied into the processInfo array using strcpy. In the case of gid and uid, the extractFirst function is implemented since this row contains three repeating numbers and only one is needed. The extractFirst function uses the strtok function to split a string delimited by tabs and returns the first instance of the uid/gid number. When all the data is present in processInfo, it is printed to the stdout for each process and the file is closed.