

Informe laboratorio 1

Julio mario monterrosa paternina c.c 1038123206

Git: https://github.com/JulioMarioUdeA/lab1_20212.git

En el enlace a git hay evidencia de que el trabajo se hizo por etapas.

Todo comienza en el main, en donde hay 'if' anidados que hacen comprobaciones para avisar de errores: si hay solo un argumento significa que es error puesto que psinfo tiene más argumentos que su nombre; de lo contrario, si hay mas de un argumento significa que el segundo es un '-l' o un '-r', si no es así es un error; en el mismo caso, si hay mas de un argumento, deben haber mas de dos argumentos, puesto que 'psinfo -l' y 'psinfo -r' reciben por lo menos un argumento, hasta ahí llega la parte de comprobación de errores por argumentos.

Para funcionar, psinfo utiliza varios archivos, un '.c' y un '.h', en el '.h' se encuentran la declaración de los métodos necesarios y en el '.c' la resolución de estos métodos:

El método 'pillarDatos' recolecta e imprime en salida estándar o en la salida deseada, los 8 datos que nos interesan del fichero 'status' que está dentro de la carpeta '/proc/<PID>'. Recibe el nombre del fichero correspondiente al proceso que deseamos averiguar, que es por ejemplo '2233', a ese nombre se le concatena por delante '/proc/' y por detrás '/status' y de esa manera se hace una ruta, con la ruta se puede abrir un stream con 'fopen' para leer el archivo línea a línea con 'getline' y por cada línea leída hacer un 'switch case' con solo 'if' preguntando con el método 'strstr' si esa línea contiene alguno de los datos que nos interesan, por ejemplo si la línea contiene 'State' significa que nos sirve, entonces esa línea se imprime a la salida que le hayamos pasado al método 'pillarDatos' como segundo parámetro, si le pasamos un NULL entonces se imprime a la salida estándar.

El método 'numeroLetras' recibe un arreglo de strings (cadena de caracteres) y el tamaño de este arreglo. Lo que hace es recorrer el arreglo y extraer de cada posición el numero de letras con 'strlen', lo pone en una variable sumadora y devuelve esta variable. Por ejemplo, para el arreglo ["hola", "paola"] va a devolver el numero 9, ya que no cuenta el carácter de fin de línea de ninguna de las posiciones.

El método 'getNombreReporte' construye el nombre del reporte que debe tener nuestro archivo de salida en caso de que usemos 'psinfo -r'. Recibe un arreglo que contiene la lista de

nombre de carpetas PID correspondiente a los procesos que se quieren averiguar por ejemplo ["22","33","3345"], también recibe su tamaño y por último un apuntador char* en donde va a quedar el nombre. El método copia al apuntador la secuencia 'psinfo-report'. Después el método recorre cada item dentro del arreglo y le extrae el nombre y le concatena '-' por delante, y se lo concatena uno a uno al apuntador que ya tiene escrito 'psinfo-report'. Por último, le concatena la secuencia '.info' para tener el nombre del reporte. Para el ejemplo anterior el nombre sería: "psinfo-report-22-33-3345.info".

Por último, está el método 'getInfoProcesos' que sirve cuando hay varios procesos a los que se les quiere averiguar los datos. Recibe el arreglo de nombre de los procesos, su tamaño y un stream. Recorre uno a uno el arreglo y a cada uno le aplica el método 'pillarDatos' detallado anteriormente. Así se obtienen los datos de muchos procesos pasados por ejemplo en el comando 'psinfo -l 22 33 3345'. El stream que se le pasa es porque 'pillarDatos' lo necesita. Este método se usa cuando el comando 'psinfo' se usa con '-l' y se le pasa un NULL como stream para que imprima los datos en pantalla, también se usa con '-r' y se le pasa el nombre del archivo de salida para imprimir los datos en dicho archivo.