|  |
| --- |
| #include <stdio.h> |
|  | #include <stdlib.h> |
|  | #include <sys/types.h> |
|  | #include <sys/stat.h> |
|  | #include <unistd.h> |
|  | #include <fcntl.h> |
|  | #include <string.h> |
|  |  |
|  | int main() |
|  | { |
|  | puts("\n\tClient - Listening\n"); |
|  | int code1 = mkfifo("file1.txt", 0666); |
|  | int code2 = mkfifo("file2.txt", 0666); |
|  | char strMessage[5000]; |
|  | if (code1 == -1) |
|  | perror("\n\tmkfile1 returned an error-file any already exist\n"); |
|  | if (code2 == -1) |
|  | perror("\n\tmkfile2 returned an error-file any already exist\n"); |
|  |  |
|  | int fd = open("file1.txt", O\_RDONLY); |
|  | int fd2 = open("file2.txt", O\_WRONLY); |
|  | if (fd == -1) |
|  | { |
|  | perror("Cannot open file1 for read"); |
|  | return EXIT\_FAILURE; |
|  | } |
|  | if (fd2 == -1) |
|  | { |
|  | perror("Cannot open file2 for write"); |
|  | return EXIT\_FAILURE; |
|  | } |
|  | puts("FIFO OPEN"); |
|  |  |
|  | char stringBuffer[5000]; |
|  | memset(stringBuffer, 0, 5000); |
|  |  |
|  | int res; |
|  | char Len; |
|  |  |
|  | { |
|  | res = read(fd, &Len, 1); |
|  |  |
|  | read(fd, stringBuffer, Len); // Read String Characters |
|  | stringBuffer[(int)Len] = 0; |
|  | printf("\nClient Received: %s\n", stringBuffer); |
|  | int j = 0, w = 0, line = 0; |
|  | while (stringBuffer[j] != '\0') |
|  | { |
|  | char ch = stringBuffer[j]; |
|  | if ((ch == ' ') || (ch == '\n')) |
|  | { |
|  | w++; |
|  | if (ch == '\n') |
|  | line++; |
|  | } |
|  | j++; |
|  | } |
|  | char LC = (char)strlen(strMessage); |
|  | char str1[256]; |
|  | char str2[256]; |
|  | char str3[256]; |
|  | sprintf(str1, " No.of Words : %d:::", w); |
|  | strcat(strMessage, str1); |
|  | sprintf(str2, " No.of Charecters: %d:::", (j - 1)); |
|  | strcat(strMessage, str2); |
|  | sprintf(str3, " No.of Lines: %d", line); |
|  | strcat(strMessage, str3); |
|  | strcat(strMessage, "\0"); |
|  | printf("\n\tString: %s", strMessage); |
|  | write(fd2, &LC, 1); |
|  | write(fd2, strMessage, strlen(strMessage)); |
|  | fflush(stdin); |
|  | strMessage[0] = 0; |
|  | } |
|  | printf("\n"); |
|  | puts("CLIENT CLOSED"); |
|  | puts("SERVER CLOSED"); |
|  | close(fd); |
|  | close(fd2); |
|  | return 0; |
|  | } |