Code:

#include<fcntl.h>

#include<sys/stat.h>

#include<string.h>

#include<sys/time.h>

#include<time.h>

#include<stdlib.h>

#include<unistd.h>

#include<errno.h>

#include<stdio.h>

int GetSize(char data[]){

int c = 0;

while(data[c] != '\0'){

c++;

}

return c;

}

void ConsoleWriteLine(char data[]){

write(1, "\n", 1);

write(1, data, GetSize(data));

}

struct timespec GetTimeSpec(int clockId){

struct timespec tspec;

switch (clockId)

{

case 1:

clock\_gettime(CLOCK\_REALTIME, &tspec);

break;

case 2:

clock\_gettime(CLOCK\_MONOTONIC, &tspec);

break;

case 3:

clock\_gettime(CLOCK\_PROCESS\_CPUTIME\_ID, &tspec);

break;

case 4:

clock\_gettime(CLOCK\_REALTIME, &tspec);

break;

default:

ConsoleWriteLine("Invalid clock identifier!!");

break;

}

return tspec;

}

void PrintLocalTime(struct timespec tspec){

struct tm \*t = malloc(sizeof(struct tm));

t = localtime(&tspec.tv\_sec);

char stringTime[100];

int size = strftime(stringTime, 100, "%c", t);

//write(1, "\n", 1);

write(1, stringTime, size);

}

void PrintTimeStamps(char path[]){

struct stat s;

stat(path, &s);

ConsoleWriteLine("Last Access Time of the file : ");

PrintLocalTime(s.st\_atim);

ConsoleWriteLine("Last Modified Time of the file : ");

PrintLocalTime(s.st\_mtim);

ConsoleWriteLine("Last Status Change Time of the file : ");

PrintLocalTime(s.st\_ctim);

}

void main(int argc, char \*argv[]){

if(argc < 3){

ConsoleWriteLine("Please Enter Operation (futimens, utimensat, utimes) and Clock type (1 to 4)!!");

}

else{

struct timespec \*current\_tspec = malloc(sizeof(struct timespec));

\*current\_tspec = GetTimeSpec(atoi(argv[2]));

struct timeval \*current\_tval = malloc(sizeof(struct timeval));

gettimeofday(current\_tval, NULL);

if(strcmp(argv[1], "futimens") == 0){

if(argc < 4){

ConsoleWriteLine("Please enter the file path!!");

}

else{

ConsoleWriteLine("Stats before futimens : ");

PrintTimeStamps(argv[3]);

int fd = open(argv[3], O\_RDWR);

perror("open");

int result = futimens(fd, current\_tspec);

if(result == -1){

ConsoleWriteLine("Error in updating the stats!!");

}

else{

perror("futimens");

ConsoleWriteLine("Stats after futimens : ");

PrintTimeStamps(argv[3]);

}

}

}

else if(strcmp(argv[1], "utimensat") == 0){

if(argc < 4){

ConsoleWriteLine("Please enter file name!!");

}

else{

int dfd = open("Test", O\_DIRECTORY | O\_RDONLY);

int result = utimensat(dfd, argv[3], current\_tspec, 0);

if(result == -1){

ConsoleWriteLine("Error in updating the stats");

}

else{

ConsoleWriteLine("Stats updated successfully!!!");

}

}

}

else if(strcmp(argv[1], "utimes") == 0){

if(argc < 4){

ConsoleWriteLine("Please enter the file path!!");

}

else{

ConsoleWriteLine("Stats before utimes : ");

PrintTimeStamps(argv[3]);

int result = utimes(argv[3], current\_tval);

if(result == -1){

ConsoleWriteLine("Error in updating the stats!!");

}

else{

ConsoleWriteLine("Stats after utimes : ");

PrintTimeStamps(argv[3]);

}

}

}

else{

ConsoleWriteLine("Wrong operation type arguement!!");

}

}

}

Output:

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated