COSC 350 System Programming

Mini Test #2 (submit to cosc350@gmail.com)

02/23/21

Name: \_\_\_\_Devin Schmidt\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (4 pt.) Write a C program which open a file (**foo**) as a input and write into a file (**symmetry**) as a symmetry of **foo** ( created output file mode will be rw------- )by using the lseek system call (do not use pread() or pwrite()).

Ex)

If content of “foo” is a string **abcdefg**, content of “symmetry” will be **gfedcba || abcdefg**.

**#include <unistd.h>**

**#include <fcntl.h>**

**#include <stdlib.h>**

**#include <stdio.h>**

**int main()**

**{**

**int inputFile, outputFile ;**

**char buffer;**

**inputFile = open("foo.txt", O\_RDONLY);**

**umask(0);**

**outputFile = open("symmetry.txt", O\_WRONLY|O\_CREAT, S\_IRUSR|S\_IWUSR);**

**while(read(inputFile, & buffer, 1) == 1){**

**write(outputFile, & buffer, 1);**

**}**

**lseek(outputFile, 0, SEEK\_END);**

**while(read(inputFile, &buffer, 1) == 1){**

**write(outputFile, &buffer, 1);**

**}**

**int fileSize = lseek(inputFile, -1, SEEK\_END);**

**while(fileSize > 0){**

**read(inputFile, &buffer, 1);**

**write(outputFile, &buffer, 1);**

**lseek(inputFile, -2, SEEK\_CUR);**

**fileSize--;**

**}**

**close (inputFile);**

**close (outputFile);**

**exit (0);**

**}**

1. (4 pt. ) Write a C program named “**sum.c**” that reads sequence of integers on the command line and prints their sum of even numbers and sum of odd number to the screen. Exit the program with an appropriate error message under the following error condition: if there is any integer on the command line.

Define your own function (named **st\_to\_int**) to change a c-string to integer. (Do not use atoi() library function)

Ex) for ./sumup 25 51 44 13 100

Output: The sum of even numbers is 231

The sum of odd numbers is 144

#include <stdio.h>

#include <stdlib.h>

int st\_to\_int(char \*str){

int result;

int x;

result = 0;

x = 1;

while ((\*str >= '0') && (\*str <= '9'))

{

result = (result \* 10) + ((\*str) - '0');

str++;

}

return (result \* x);

}

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

{

int sumEven;

int sumOdd;

if(argc<2){

exit(0);

}

else{

for(int i; i<argc; i++){

if(argv[i]%2 == 0){

sumEven = sumEven + st\_to\_int(argv[i]);

}

if(argv[i]%2 !=0){

sumOdd = sumOdd + st\_to\_int(argv[i]);

}

}

}

printf("The sum of even numbers are %d", sumEven);

printf("The sum of odd numbers are %d", sumOdd);

exit (0);

}

1. (1 pt.)Probably the most important issue in implementing file storage is keeping track of which disk blocks go with which file. Various methods are used in different operating systems: contiguous allocation, linked-list allocation, linked-list with FAT and index-node. Briefly discuss each idea.

* Contiguous allocation – a file saved in continuous blocks
* Linked-list allocation – a file saved in blocks and
* Linked-list allocation with FAT(file allocation table) – Stored in main memory but can take up a lot of memory
* Index-Node – stored with the block addresses and attributes

1. (1 pt.) Write a C program that reads a file name on the command line and prints the file size by bytes. Do not use lseek(), stat(), fstat(), lstat(), pread(), pwrite() system call. Do not use any bash command by using system() system call.

#include <stdio.h>

#include<stdlib.h>

#include<fcntl.h>

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

char buffer;

int fileSize = 0;

int file = open(argv[1], O\_RDONLY);

while(read(file, &buffer, 1) > 0){

fileSize++;

}

printf("%d",fileSize);

}