***Assignment-1***

***CS-312***

Task 1: (Code)

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <string.h>

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

char \*input\_filename = argv[1];

char \*output\_filename = argv[2];

char c;

size\_t nbytes = 4096; // buffer size

if (argc != 3) {

fprintf(stderr, "Usage: %s <input-filename> <output-filename>\n", argv[0]);

return 1;

}

int in, out;

in = open(input\_filename, O\_RDONLY);

if (in == -1) {

fprintf(stderr, "Error opening input file 1: %s\n", input\_filename);

return 3;

}

out = open(output\_filename, O\_WRONLY|O\_CREAT, S\_IRUSR|S\_IWUSR);

if (out == -1) {

fprintf(stderr, "Error opening input file 2: %s\n", output\_filename);

close(in);

return 4;

}

while(read(in,&c,nbytes) == 1){

write(out,&c,nbytes);

}

exit(0);

}

A: 600MB

Task 2:

|  |  |  |
| --- | --- | --- |
| Buffer Size  (Bytes) | Total Time  (seconds) | System Time  (seconds) |
| 4096 |  |  |
| 8192 |  |  |
| 16384 |  |  |

B: 1200MB

|  |  |  |
| --- | --- | --- |
| Buffer Size  (Bytes) | Total Time  (seconds) | System Time  (seconds) |
| 4096 |  |  |
| 8192 |  |  |
| 16384 |  |  |

|  |  |  |
| --- | --- | --- |
| Buffer Size  (Bytes) | Total Time  (seconds) | System Time  (seconds) |
| 4096 | 0.004 | 0.004 |
| 8192 | 0.003 | 0.003 |
| 16384 | 0.005 | 0.005 |

|  |  |  |
| --- | --- | --- |
| Buffer Size  (Bytes) | Total Time  (seconds) | System Time  (seconds) |
| 4096 |  |  |
| 8192 |  |  |
| 16384 |  |  |

|  |  |  |
| --- | --- | --- |
| Buffer Size  (Bytes) | Total Time  (seconds) | System Time  (seconds) |
| 4096 | 0.005 | 0.005 |
| 8192 | 0.006 | 0.006 |
| 16384 | 0.005 | 0.005 |

C: Total Time

|  |  |  |
| --- | --- | --- |
| Buffer Size  (Bytes) | Total Time (seconds) | |
| File size  (600MB) | File size  (1200MB) |
| 4096 | 0.010 | 0.008 |
| 8192 | 0.012 | 0.006 |
| 16384 | 0.010 | 0.010 |

D: How does the total-time vary as a function of buffer-size?

Ans: Increasing the buffer size can lead to faster file transfers up to a certain point. After that point, the benefits of increasing the buffer size begin to plateau, and larger buffer sizes may not result in any further speed improvements. The optimal buffer size can vary depending on the specific system and file being transferred, and may require some experimentation to determine.

Task 3:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/wait.h>

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

if (argc != 3) {

fprintf(stderr, "Usage: %s file1 file2\n", argv[0]);

exit(1);

}

int status;

pid\_t pid = fork();

if (pid < 0) {

fprintf(stderr, "Error forking child process\n");

exit(2);

}

else if (pid == 0) { // Child process

execlp("cmp", "cmp", "-s", argv[1], argv[2], (char \*) NULL);

perror("Error executing child process");

exit(2);

}

else { // Parent process

if (waitpid(pid, &status, 0) == -1) {

fprintf(stderr, "Error waiting for child process\n");

exit(2);

}

if (WIFEXITED(status)) {

int exit\_code = WEXITSTATUS(status);

if (exit\_code == 0) {

printf("Files %s and %s are the same\n", argv[1], argv[2]);

}

else if (exit\_code == 1) {

printf("Files %s and %s are different\n", argv[1], argv[2]);

}

else {

fprintf(stderr, "Error opening files %s and %s\n", argv[1], argv[2]);

exit(2);

}

}

else if (WIFSIGNALED(status)) {

fprintf(stderr, "Child process terminated abnormally\n");

exit(2);

}

}

return 0;

}