Homework 10

Group Number 56 | Anmol Singhal 2017332, Daksh Shah 2017336, Tejas Oberoi 2017367

The output of original code:
./a.out 0.74s user 1.73s system 98% cpu 2.499 total

Modified Code-

```c
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <sys/mman.h>

#define NUM_PAGES 1024
#define PAGE_SIZE 4096

void write_file(int fd, char * buf) {
    int i, ret;

    for (i = 0; i < NUM_PAGES; i++) {
        ret = write(fd, buf, PAGE_SIZE);
        if (ret != PAGE_SIZE) {
            printf("unable to write\n");
            exit(0);
        }
    }
}

void read_file(void * mappedMemory, char * buf) {
    int i, ret;

    for (i = 0; i < NUM_PAGES; i++) {
        if (memcmp((((char*)mappedMemory)+i*PAGE_SIZE, buf, PAGE_SIZE) != 0)
{
            printf("data don't match\n");
            exit(0);
        }
```

```c
        }
}

int main() {
    int i, fd, ret;
    char buf[PAGE_SIZE];

    for (i = 0; i < PAGE_SIZE; i++)
        buf[i] = rand() % 128;
    fd = creat("/tmp/temp.txt", S_IRUSR | S_IWUSR);
    if (fd < 0) {
        printf("error in creat\n");
        exit(0);
    }
    write_file(fd, buf);
    close(fd);

    fd = open("/tmp/temp.txt", O_RDONLY);

    //Mapping the File to Memory
    //     mmap(void *addr, size_t len, int prot, int flags, int fd, off_t
offset);
    void * mappedMemory = mmap(NULL, NUM_PAGES*PAGE_SIZE, PROT_READ,
MAP_SHARED, fd, 0);
    if(mappedMemory == (void *) -1)
        exit(1);

    for (i = 0; i < 1024; i++)
        read_file(mappedMemory, buf);

    //unmap: Releasing the memory
    munmap (mappedMemory, NUM_PAGES*PAGE_SIZE);

    close(fd);
    return 0;
}
```

The output of modified code (after using mmap):
./a.out 0.35s user 0.02s system 97% cpu 0.381 total

```
● ● ●   1. daksh@Dakshs-MacBook-Air: ~/Desktop/IIITD/Operating Systems/os-group-hws/Homework 10 (zs...
~/Desktop/IIITD/Operating Systems/os-group-hws/Homework 10git:(master) gcc -O3 original.c
~/Desktop/IIITD/Operating Systems/os-group-hws/Homework 10git:(master) time ./a.out
./a.out  0.74s user 1.73s system 98% cpu 2.499 total
~/Desktop/IIITD/Operating Systems/os-group-hws/Homework 10git:(master) gcc -O3 source.c
~/Desktop/IIITD/Operating Systems/os-group-hws/Homework 10git:(master) time ./a.out
./a.out  0.35s user 0.02s system 97% cpu 0.381 total
~/Desktop/IIITD/Operating Systems/os-group-hws/Homework 10git:(master) ▊
```

Explanation: Time spent with mmap is 0.381 seconds, while without mmap is 2.499 seconds. The reason behind it is the overhead of reading data from disk (hard disk). In the normal execution, the file was reading the file from the disk 1024 times. Since each time the data was fetched from the disk, the delay caused by the hard disk magnified. On the other hand, when we used mmap, it read the file from disk exactly once and mapped it to the main memory (RAM). RAM has a much faster access time than hard disk and hence when we read the file 1024 times from the RAM, it took much lesser time.