Homework 10: mmap

The goal of this homework is to understand the mmap system call. The program in Section 1 creates a file and writes 4 MB data to it. After this step, the program opens the file in read-only mode and sequentially read the entire file 1024 times.

1 Program

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.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 (int fd, char *buf)
{
    int i, ret;
    char str[PAGE_SIZE];
    lseek (fd, 0, SEEK_SET);

    for (i = 0; i < NUM_PAGES; i++) {
        ret = read (fd, str, PAGE_SIZE);
        if (ret != PAGE_SIZE) {
            printf ("unable to read %d\n", ret);
            exit (0);
        }
        if (memcmp (str, buf, PAGE_SIZE) != 0) {
            printf ("data don't match\n");
            exit (0);
        }
    }
}

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);
    for (i = 0; i < 1024; i++) {
        read_file (fd, buf);
    }
    close (fd);
    return 0;
}
```

2 Turn in

- Compile the program using: `‘‘gcc -O3 filename.c’’`
- Run the program using: `time ./a.out`
- Write the output of the above command.

The time command prints total time, time spent in user mode, and time spent in kernel mode. Modify, the program to use mmap (look at the man page of mmap) system call to map the file in read mode after the data is stored in the file (using write system call). Modify `read_file` to reading from memory mapped file instead of using the read system call. Finally, unmap the file before closing the file descriptor.

- Compile you modified program using `‘‘gcc -O3 filename.c’’`
- Run the program using: `time ./a.out`
- Write the output of the above command.
- Explain, why time spent in the kernel are different with and without `mmap`?

3 Submission

Upload a pdf file with your answers on the Backpack. This is a project group homework. Only one project member needs to upload. You must follow the naming convention as group_id.pdf.