CS 342 Operating Systems
Spring 2024
Project 4

Murat Çağrı Kara - 22102505
Görkem Kadir Solun - 22003214

fatmod DISKIMAGE -l

Number of Files Created Listing vs Command Execution Time

In this experiment, we studied how the execution time of the `fatmod DISKIMAGE -l` command is affected by the number of files in the root directory. We created a root directory containing different numbers of files and measured the time it takes for the command to run. The results of this experiment are summarized in a table:

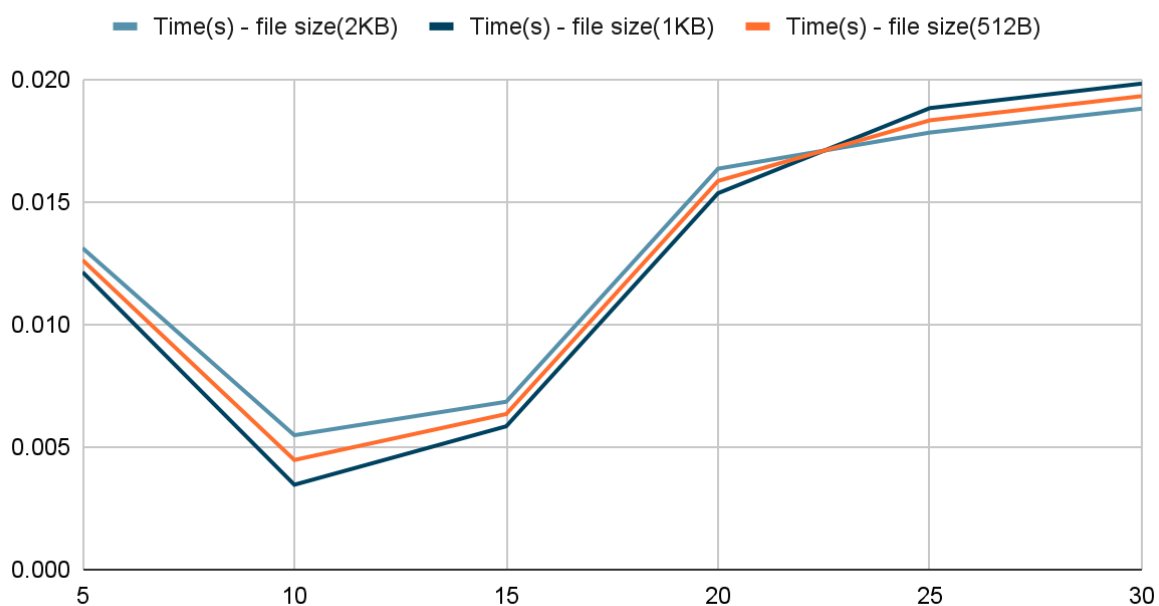## # of Files vs Time (s) listing files command

# of Files (2KB)
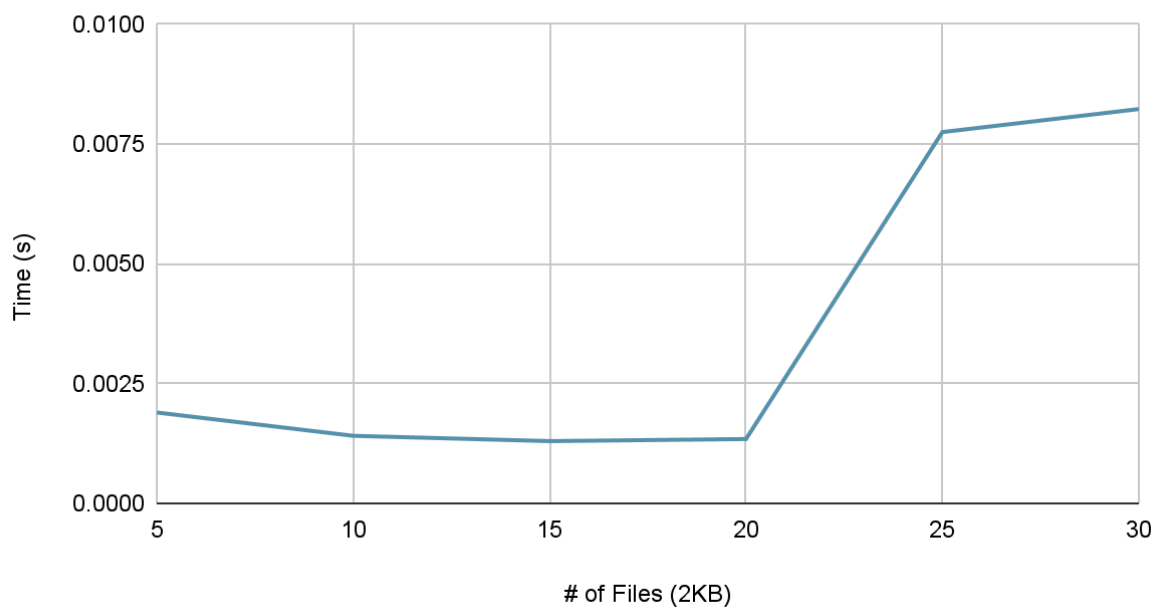
The table demonstrates a clear positive correlation between the number of files in the FAT32 disk image's root directory and the execution time of the `fatmod DISKIMAGE -l` command. This confirms that as the number of files increases, the execution time of the listing files command also increases.

fatmod DISKIMAGE -r -a FILENAME

File Size Length and Number of Files vs Command Execution Time

This experiment analyzes how the `fatmod DISKIMAGE -r -a FILENAME` command's performance, measured by execution time, changes with increasing file sizes as well as increased number of files. We tested the read command on files of sizes 2KB, 1KB and 512B, as well as different numbers of files for each of these file sizes and recorded the time taken for each execution. The results of this experiment are presented in a table:

of Files vs Execution Time

Time(s) - file size(2KB) — Time(s) - file size(1KB) — Time(s) - file size(512B)

The graph illustrates a positive correlation between the size of the files and the execution time of the `fatmod DISKIMAGE -r -a FILENAME` command generally. Moreover, demonstrates a clear positive correlation between the number of files in the FAT32 disk image's root directory and reading the content inside of them. The fluctuation at the beginning can be caused by the system having other tasks running concurrently.
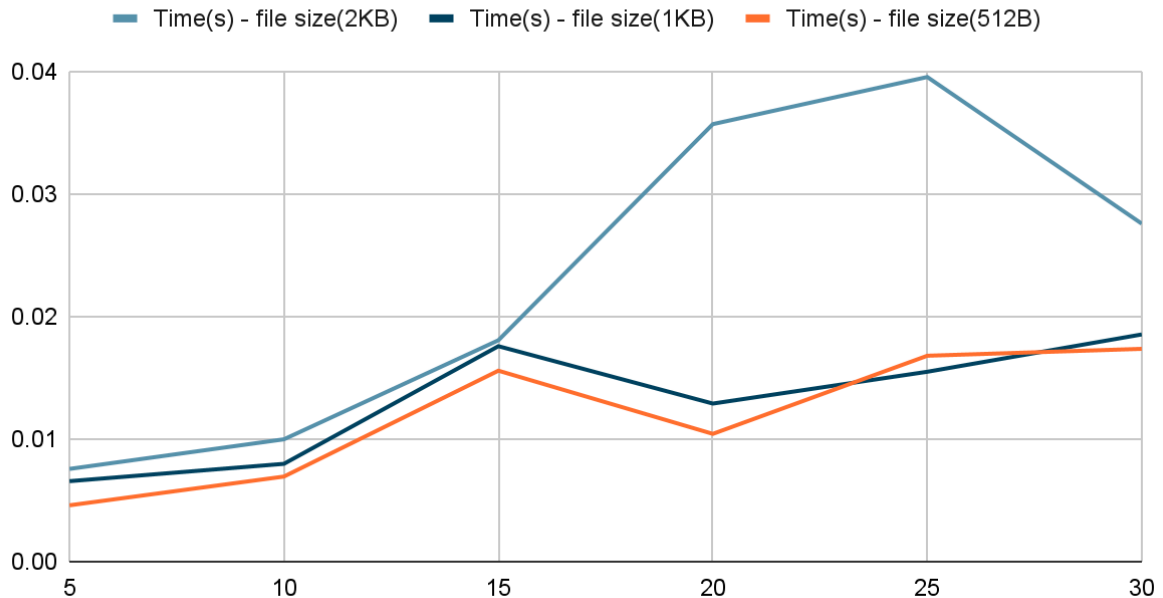
fatmod DISKIMAGE -r -b FILENAME

File Size Length and Number of Files vs Command Execution Time

This experiment analyzes how the `fatmod DISKIMAGE -r -b FILENAME` command's performance, measured by execution time, changes with increasing file sizes as well as increased number of files. We tested the read binary command on files of sizes 2KB, 1KB and

512B, as well as different numbers of files for each of these file sizes and recorded the time taken for each execution. The results of this experiment are presented in a table:
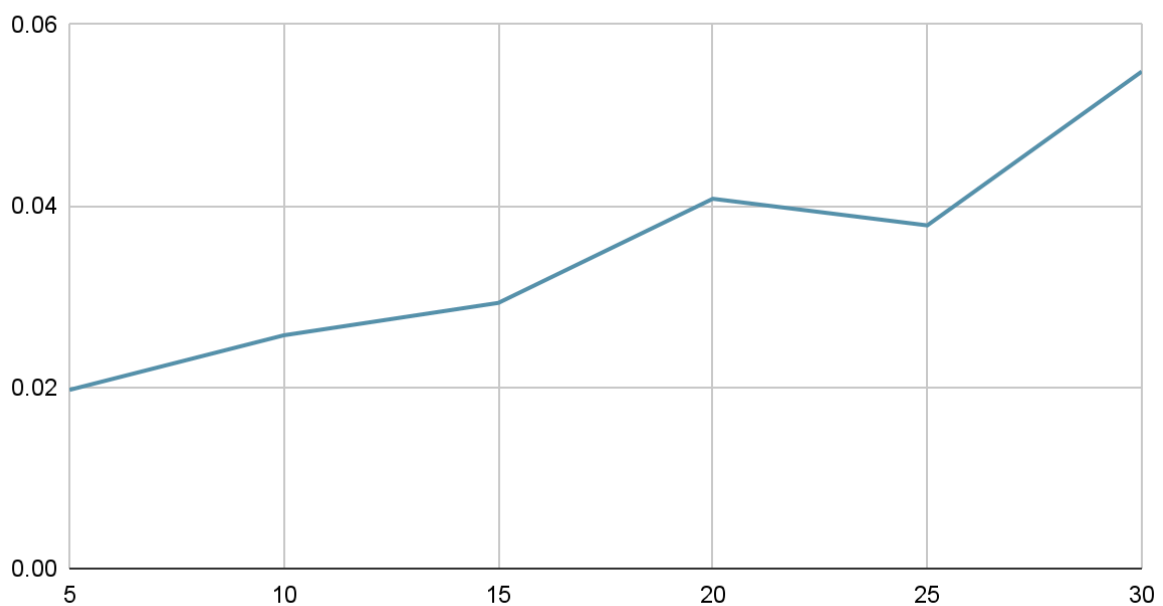
of Files vs Execution Time

The graph clearly illustrates a positive correlation between the size of the files and the execution time of the `fatmod DISKIMAGE -r -b FILENAME` command. Moreover, demonstrates a clear positive correlation between the number of files in the FAT32 disk image's root directory and reading the content as binary inside of them.

fatmod DISKIMAGE -c FILENAME

Number of Files Created vs Command Execution Time

This experiment examines the impact of the number of files in the root directory on the performance (execution time) of the `fatmod DISKIMAGE -c FILENAME` command. We created a root directory containing different numbers of files and measured the time it took for the file creation command to run. The results of this experiment are presented in the table:
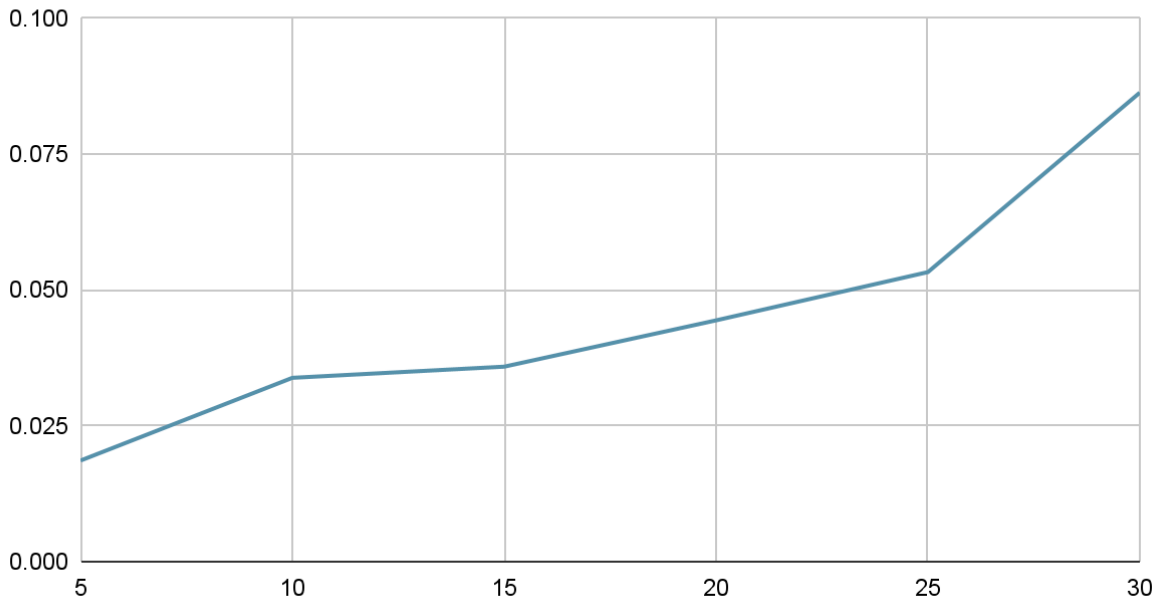

# of Files vs Execution Time

fatmod DISKIMAGE -d FILENAME

Deleting Number of Files vs Command Execution Time

This experiment explores the relationship between the number of files in the root directory and the execution time of the `fatmod DISKIMAGE -d FILENAME` command. We created a root directory with varying file counts and measured the time it took to execute the file deletion command for each directory. The results are presented in a table:

of Files vs Execution Time

The graph clearly shows a positive correlation between the number of entries in the FAT32 disk image's root directory and the execution time of the `fatmod DISKIMAGE -d FILENAME` command. This confirms that as the number of files increases, the time it takes for the deleting file command to execute also increases.

fatmod DISKIMAGE -w FILENAME OFFSET N DATA

Number of Files and Number of File Sizes vs Command Execution Time

This experiment explores how the execution time of the `fatmod DISKIMAGE -w FILENAME OFFSET N DATA` command changes with an increasing number of files in the root directory. We created a root directory with different file counts and measured the time it takes to execute the write command. Moreover, this experiment investigates how the execution time of the `fatmod DISKIMAGE -w FILENAME OFFSET N DATA` command changes with an increasing number of characters to be written (represented by the `N` parameter). We tested the command with varying `N` values (2KB, 1KB and 512B) and measured the time it takes for each execution. The results are presented in a table:

of Files vs Execution Time

— Time(s) - file size(2KB)   — Time(s) - file size(1KB)   — Time(s) - file size(512B)
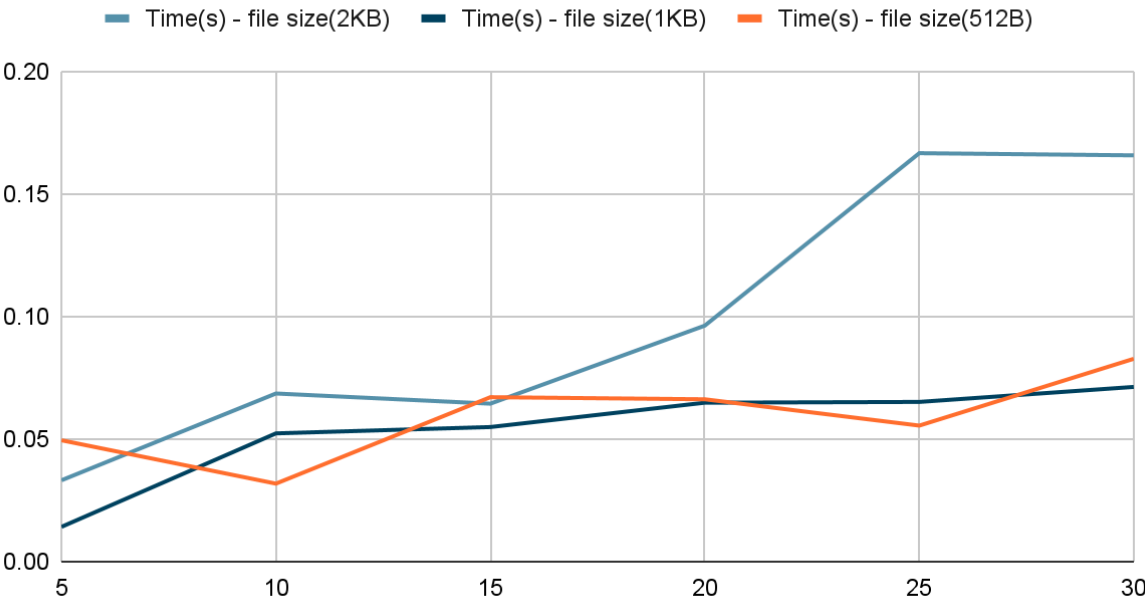
Conclusion

This study investigated how the performance of `fatmod DISKIMAGE` commands is affected by two key factors: the number of files and the size of data being processed.

Firstly, as the number of files in the root directory increases, all `fatmod` commands, regardless of whether they are reading or writing data, experience a slowdown in execution time.

Secondly, the size of the files themselves also plays a role. The `fatmod DISKIMAGE -r` (read) command takes longer to execute when dealing with larger files, both in binary and ASCII formats. This is understandable, as the tool needs to transfer a greater amount of data during the read operation. Similarly, the `fatmod DISKIMAGE -w` (write) command's performance is impacted by the number of characters to be written (represented by the `N` parameter). A larger `N` value translates to more data being written to the disk image, which consequently increases the execution time.

In conclusion, both the quantity of files and the size of data being handled influence the execution speed of `fatmod DISKIMAGE` commands.