

COMP 3700: Project 2

auDiskTool - A Tool for Monitoring Disk Devices

Points: (100 points) -- **Deadline: 11:59pm Feb 18th, 2024**



1. Building your Linux disk monitoring tool - auDiskTool

In this project, you have a Linux tool called – auDiskTool – to monitor disk performance (i.e., I/O transfer rates) in a Linux system. Your auDiskTool can output a file containing a list of reports (attached at the end) that help system administrators in configuring the Linux system to achieve good disk I/O performance.

The monitoring reports created and maintained by your auDiskTool offers statistical information on I/O transfer rates. Our long-term goal is building a tool to monitor both processor and disk performance of Linux systems. However, the short-term goal to be achieved in this project is to generate a state diagram. After you complete this project, you may extend your auDiskTool to monitor processor performance in addition to disk performance.

2. Requirements

2.1. Statistical Information

Each report item recorded in an output file contains statistics on a per disk or partition basis. auDiskTool allows users to choose a particular disk or partition to monitor. If no disk nor partition is chosen by the users, then auDiskTool monitors all disks used by the Linux system.

Each report item may show the following information:

Device: This column provides the disk or partition name (e.g., sda). If your Linux machine has new kernels, the device name listed in the /dev directory is displayed. If your Linux kernel is 2.4, the format of the names is devm-n, where **m** is the major number of the device, and **n** is a

distinctive number. If your Linux kernel is 2.2 (this is an uncommon case), the name will be displayed as hdiskn.

Blk_read:

This column reports the total number of reads.

Blk_read/s:

This column indicates the number of reads from the disk per second.

KB_read/s:

This column indicates the amount of data blocks read from the disk per second (i.e., measured in Kilobytes per second). **Note:** we assume the size of sector is 1 KB.

Blk_wrtn:

This column reports the total number of blocks written.

KB_wrtn/s:

This column indicates the number of data blocks written to the disk per second. (i.e., measured in Kilobytes per second.)

Blk_wrtn/s:

This column indicates the number of data blocks written to the disk per second.

2.2. Configure Time Interval and Count

Your auDiskTool can set an interval parameter, which specifies the amount of time measured in seconds between two consecutive disk reports. Each disk report contains statistics collected during the interval since the previous report is recorded. Your auDiskTool also can decide the number of reports to be collected in an output file. If the interval parameter and the count parameter are not specified, auDiskTool may use default values. For example, the default value of the interval parameters is 1 second; the default value of the count parameter is 10.

2.3. Specify the File Name of an Output Report

Your auDiskTool should allow users to specify the file name of an output report. The file name may be fully specified with a path. If no path is provided, then the working directory will be the current directory where the new report file is created. If the output file exists, then new report items will be appended at the end of the existing file.

Note 1: If users specify a file name that does exist, auDiskTool must inform users that an output report file with the same name exists and reported items will be added to the existing file.

Note 2: If the report file name is not specified, then “report.adt” will be used as a default output file name.

2.4. Specify what statistical data to be reported

In section 3.1, we list 7 statistical data items. Your tool should allow users to decide what data items to be included in a report. A configuration file (see Section 3.5 below) stores default values for these decisions.

2.5. A Configuration File – audisktool.conf

All the default parameters (e.g., time interval, count, output file name) are stored in a configuration file. This configuration file is loaded into main memory when auDiskTool starts its execution. The configuration file name is “audisktool.conf”. The format of the configuration file is:

```
Interval, count, print_blk_read, print_blk_read/s,  
print_kb_read/s, print_blk_write, print_blk_write/s,  
print_kb_write/s
```

The values of print_blk_read, print_blk_read/s, print_kb_read/s, print_blk_write, print_blk_write/s, print_kb_write/s can be either ‘1’ or ‘0’. ‘1’ means that the value will be reported; ‘0’ means the value is ignored in the report.

For example, suppose we have the following configuration file:

```
5 10 1 1 1 0 0 0
```

The above file indicates that Interval is 5 seconds, count is 10, report values of blk_read, blk_read/s, kb_read/s and do not include the values of blk_write, blk_write/s, kb_write/s in the report.

You do not need to submit this configuration file via Canvas; the TA will use the configuration file with the following parameters to test your implementation:

```
1 10 1 1 1 1 1 1
```

2.6. Display the report

Users are allowed to open the report and display monitoring records inside audisktool. If the report file does not exist or cannot be opened, audisktool must show a warning message.

2.7. System Quit

This should safely terminate the audiskTool. If any parameter (e.g., time interval and count) is updated, the system parameters must be saved back to the configuration file called “audisktool.conf”.

3. Retrieve Disk Statistics

The Linux (version 2.4.20 and above) operating system offers extensive disk statistics to measure disk activities. You can use the following command to check the version of your Linux:

```
$uname -r
```

Linux Version 2.6 and above:

The disk statistical information can be found in

```
/proc/diskstats
```

You can use the following command to display this file:

```
$cat /proc/diskstats
```

Example 1: Below is an example to show the format of the above file:

```
3      0      sda 446216 784926 9550688 4382310 424847 312726 5922052 19310380
0 3376340 23705160
```

```
3      1      sda1 2 0 4 24 0 0 0 0 0 24 24
```

You also can use the following command to display the information related to disks in the `/proc/diskstats` file.

```
$grep 'sda' /proc/diskstats
```

Note that `grep` is a utility program for searching plain-text data sets for lines matching a regular expression (e.g., 'sda' in our case).

4. Format of “/proc/diskstats”

In example 1 shown on page 4, you can find each row has 14 items. The first three items are the major and minor device numbers, and device name. For example, given the following row:

```
3      1      sda1 2 0 4 24 0 0 0 0 0 24 24
```

The major device number is 3, and minor device number is 1, and device name is `sda1`.

The 11 fields listed after the device name are statistics data of the device whose major/minor device numbers as well as name are shown in the first three fields. All these 11 fields except field 9 are cumulative since boot. Note that field 9 goes to zero as I/Os complete; all others only increase. These fields are unsigned long numbers.

The 11 fields are explained below:

Field 1: # of reads completed. This is the total number of reads completed successfully.

Field 2/6: # of reads/writes merged. Reads and writes which are adjacent to each other may be merged for efficiency. Thus two 4K reads may become one 8K read before it is ultimately handed to the disk, and so it will be counted (and queued) as only one I/O. This field lets you know how often this was done.

Field 3: # of sectors read. This is the total number of sectors read successfully.

Field 4: # of milliseconds spent reading. This is the total number of milliseconds spent by all reads (as measured from `__make_request()` to `end_that_request_last()`).

Field 5: # of writes completed. This is the total number of writes completed successfully.

Field 7: # of sectors written. This is the total number of sectors written successfully.

Field 8: # of milliseconds spent writing. This is the total number of milliseconds spent by all writes (as measured from `__make_request()` to `end_that_request_last()`).

Field 9: # of I/Os currently in progress. The only field that should go to zero. Incremented as requests are given to appropriate struct *request_queue* and decremented as they finish.

Field 10: # of milliseconds spent doing I/Os. This field increases so long as field 9 is nonzero.

Field 11: weighted # of milliseconds spent doing I/Os. This field is incremented at each I/O start, I/O completion, I/O merge, or read of these stats by the number of I/Os in progress (field 9) times the number of milliseconds spent doing I/O since the last update of this field. This can provide an easy measure of both I/O completion time and the backlog that may be accumulating.

5. Rubrics:

1. Please list class names. (18 points)
2. Use argoUML to create a state chart diagram. (10 points)
3. For each class, please add proper info – at least one attribute and operation. If no ops, please indicate N/A-- for software engineers to define them. (16 points)
4. Please indicate the relationship among classes with proper types of lines. (20 points)
5. Please indicate the conditions for transitions. (16 points)
6. Please specify the multiplicity. (15 points)
7. Submit your solution on Canvas with the format: project2_firstname.zargo. (5 points)

6. No Late Submission

- Late submissions will not be accepted and will result in a **ZERO** without valid excuses, in which case you should talk to Dr. Li to explain your situation.
- GTA/Instructor will **NOT** accept any late submission caused by Internet latency.

7. Rebuttal period

- You will be given a period of two business days to read and respond to the comments and grades of your homework or project assignment. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.

Sample Usage

1. Help

```
./audisktool_aak0010 <- aak0010 is a TA username
auDiskTool, version 1.0.0. Type 'help' to find more about commands.
```

>help

```
run - run the monitoring tool.
set interval [value] - set sampling period to [value] set
count [value] - set the number of records to [value] set
report [name] - set report file name to [name]
set blk_read [0|1] - set print_blk_read to 0 or 1 set
blk_read/s [0|1] - set print_blk_read/s to 0 or 1 set
kb_read/s [0|1] - set print_kb_read/s to 0 or 1 set
blk_write [0|1] - set print_blk_write to 0 or 1 set
blk_write/s [0|1] - set print_blk_write/s to 0 or 1 set
kb_write [0|1] - set print_kb_write to 0 or 1 print
conf - display all the parameters
print report - open and display the report file
save - the configuration file audisktool.conf is updated
```

```
display - exit - exit the tool.
```

>

2. Run the tool

```
./audisktool_aak0010
auDiskTool, version 1.0.0. Type 'help' to find more about commands.
```

>run

```
Monitoring time = 5 Seconds, Number of records = 10, print_blk_read =
1, print_blk_read/s = 1, print_kb_read/s = 1,
print_blk_write = 0, print_blk_write/s = 0, print_kb_write/s = 0, report
file name = 'report.adt'
Please wait ...
```

```
A file "report.adt" is updated.
```

>

3. Change report file name

```
./audisktool_aak0010
auDiskTool, version 1.0.0. Type 'help' to find more about commands.
```

>set report aak0010.adt

```
The report file name is changed from 'report.adt' to 'aak0010.adt'. You
can now type 'run' to generate new records to be saved in 'aak0010.adt'.
Note: 'report.adt' will not be deleted by audisktool.
```

>

4. Display records in the report file

4.1 no record is found

```
./audisktool_aak0010
auDiskTool, version 1.0.0. Type 'help' to find more about commands.
```

```
>print report
```

```
No record found in 'report.adt'
```

```
>
```

4.2 Records are found

```
./audisktool_aak0010
auDiskTool, version 1.0.0. Type 'help' to find more about commands.
```

```
>print report
```

```
2 records found in 'report.adt'
```

blk_read	blk_read/s	kb_read/s	blk_write	blk_write/s	kb_write/s
10	1.13	2.26	N/A	N/A	N/A
55	5.4	10.8	N/A	N/A	N/A

```
>
```

5. Change count and interval

```
./audisktool_aak0010
auDiskTool, version 1.0.0. Type 'help' to find more about commands.
```

```
>set count 15
```

```
The number of records generated in each run has been changed to 15.
```

```
>set interval 3
```

```
The sampling interval has been changed to 3 seconds.
```

```
>
```

6. Change print_blk_read and other similar parameters

```
./audisktool_aak0010
auDiskTool, version 1.0.0. Type 'help' to find more about commands.
```

```
>set blk_read 0
```

```
Print_blk_read has been changed to 0.
```

```
>set blk_write 0
```

```
Print_blk_write was 0; the parameter remains unchanged to 0.
```

```
>
```

7. Display all the parameters.

```
./audisktool_aak0010
auDiskTool, version 1.0.0. Type 'help' to find more about commands.
```

```
>print conf
```

```
Monitoring time = 5 Seconds, Number of records = 10, print_blk_read =
1, print_blk_read/s = 1, print_kb_read/s = 1,
print_blk_write = 0, print_blk_write/s = 0, print_kb_write/s = 0, report
file name = 'report.adt'
```

```
>
```

8. Save the configuration file

8.1 no need to save

```
$/audisktool_aak0010
auDiskTool, version 1.0.0. Type 'help' to find more about commands.

>save
audisktool.conf has not been updated. There is no need to save the file.
```

8.2 change can save the configuration file

```
$/audisktool_aak0010
auDiskTool, version 1.0.0. Type 'help' to find more about commands.

>set blk_read 0
Print_blk_read has been changed to 0.

>save
file audisktool.conf has been updated.

>
```