|  |
| --- |
| American University in Cairo |
| File Delve: A Disk Analyzer |
| An Operating System’s Course Project |

|  |
| --- |
| 3-12-2015 |

Ahmed Abouseida 900121315

Mariam Aziz 900114864

Mahmoud Khodary 900120434

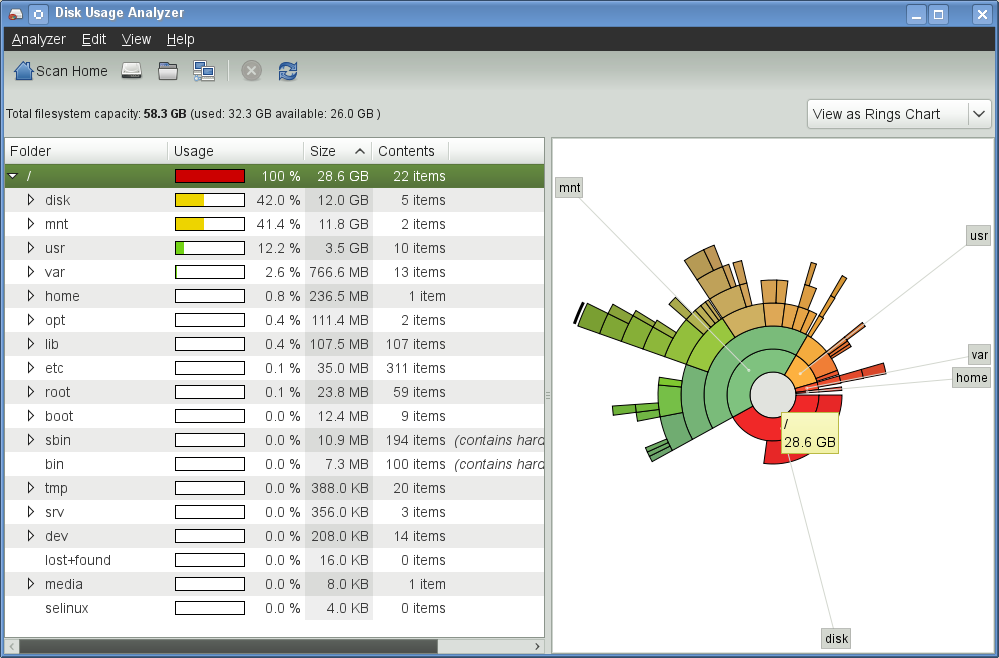
Sara Soliman 900123107

Aya Harb 900120958

# Case Study of Similar Disk Analyzers:

After reviewing multiple disk analyzer, on all three main operating systems (Linux, Mac OS, and Windows), we came up with a list of the best analyzers out there. We then compared all the analyzers and came up with the two best examples that sum up the best features of a disk analyzer. These were Baobab and JDiskReport. Both are reviewed below.

# Baobab



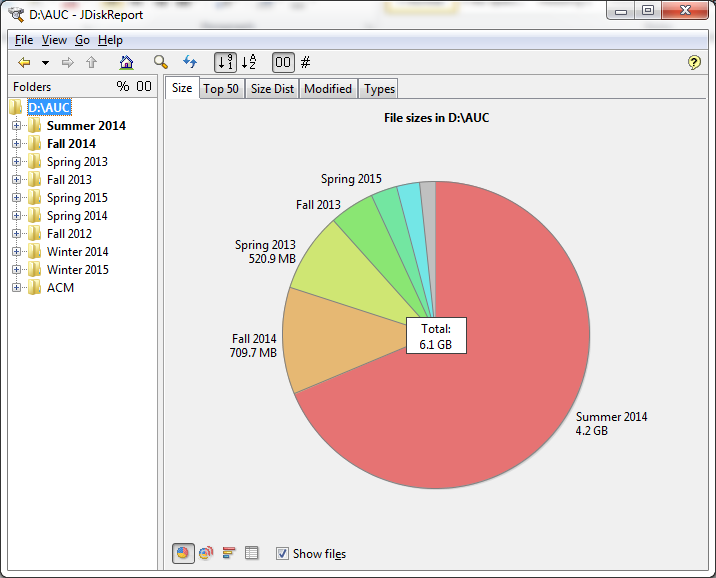
Best Features:

* Baobab offers Real-time refresh, allowing changed done to the system at the run time of the program to be portrayed on the graphs.
* Baobab also allows checking of the system as a whole of specific folders within.
* Last key feature of Baobab is it’s efficient GUI. This allowed for:
  + Rings representing different levels of the file system, allowing clear view of parents, illustrated using different sized and colors.
  + Navigating the system from both listed and donut graph view.
  + Showing the folder details in term of usage % of parent directory, the Folder name as well as the number of files contained.

Disadvantages:

* Baobab only shows directories, not files

# JDiskReport

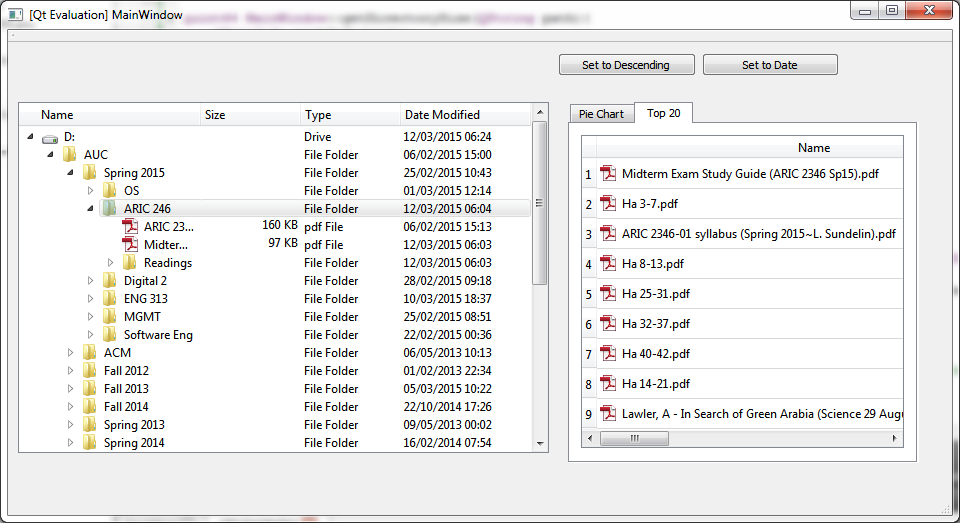
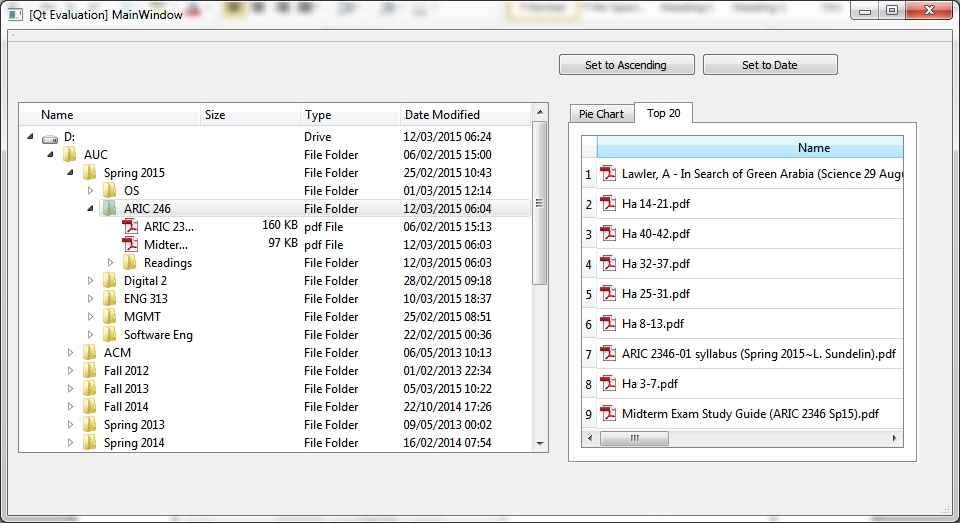


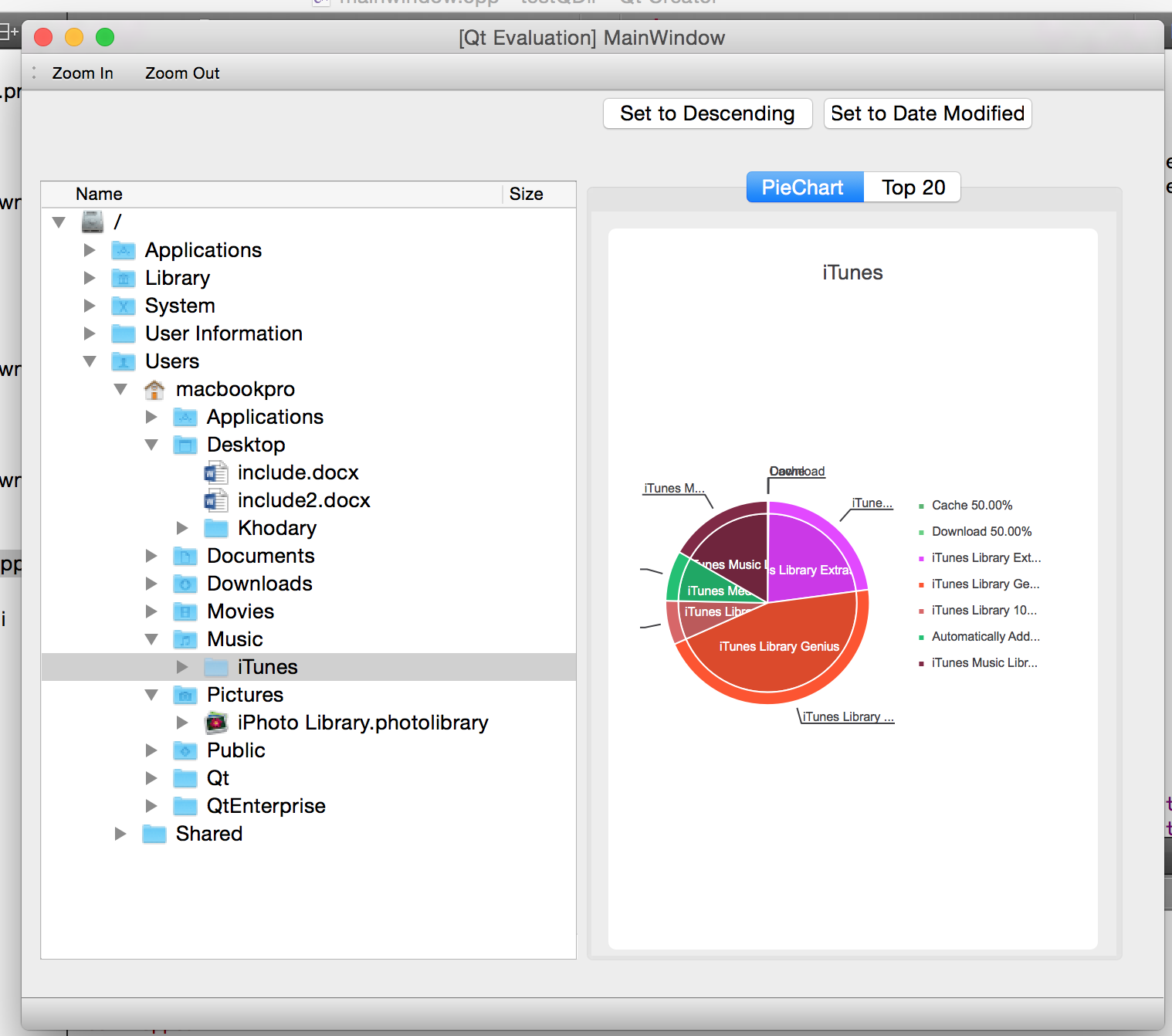
Best Features:

* JDiskReport allows checking of the system as a whole of specific folders within.
* Another key feature of JDiskReport is its efficient GUI. This allowed for:
  + Rings representing different levels of the file system, allowing clear view of parents, illustrated using different sized and colors.
  + Navigating the system from both listed and donut graph view.
  + Showing the folder details in term of usage % of parent directory, the Folder name as well as the number of files contained.
  + Availability of both a pie chart and a bar chart to make viewing easier.
  + It offers a tabular view of the data
* It offers the top 50 newest and oldest files
* Shows how sizes are distributed
* Shows how date of modification relates to sizes
* Shows the sizes in correspondence to a file time.
* It gives the option of viewing or ignoring files.

**Features:**

After reviewing the previous products, as well as factoring in the time constrain of the project, these are the features we agreed upon and implemented.

* Top 20 files, in ascending and descending orders:
* Piechart View:



**How it Works:**

The root is parsed, getting all directories and files available on disk.

These are then put into a TreeView with the appropriate details. These include the date modified and the type of the file.

Clicking on any folder in the tree created a table of the top 20 files in terms of size in ascending order.

Lastly, clicking on an item created a pie chart representation of the files and directories, with each occupying a slice of the pie chart, with the name shown.

**Work distribution:**

This project went through multiple stages of production, and thus not all files attached and mentioned below are part of the final product, however they were parents that members of the project worked on greatly! The original plan of the project was writing a disk analyzer program in c that produced a text file with all the data for the files and directories. This was then parsed by the GUI and used to produce graphs. Due to time contains, and as of the morning of the day this is being submitted, the DU.c file was replaced with a QT library, and this both the DU.c and the parser rendered useless. Although not used, these files consumed a lot of time from all members.

The work done is as follows, all that has an \* next to it has been rendered useless by the last minute change on Thursday.

**Ahmed Abouseida 900121315**

Parsing\*

Sorting

Debugger

**Mariam Aziz 900114864**

Piechart

**Mahmoud Khodary 900120434**

PieChart

QFileSystem

TreeWidget

**Sara Soliman 900123107**

Piechart

Parsing\*

Sorting

**Aya Harb 900120958**

DU.cpp\*

Report