**Operating Systems**

**CS4348**

**Project #3: Disk Allocation Methods**

**Due Date: Saturday, December 1, 2018**

## I. Project Organization

This project demonstrates three different file allocation methods.

You should do the following pieces to complete your project. Each piece is explained below:

* Code 60 points
* Output 30 points
* Summary 10 points

# Code

Your code should be nicely formatted with plenty of comments. The code should be easy to read, properly indented, employ good naming standards, good structure, etc.

# Output

Output will be graded by running your program.

# Summary

The summary section should discuss your project experience. You should discuss how you approached the project, including anything difficult or interesting, what was learned, and the end result. This should be at least 1 page in length.

## II. Project Description

**Language/Platform/Approach**

This project must target a Unix platform and execute properly on our CS1 server.

The project must be written in C++ or Java.

You should use the approach described until Details unless you have obtained instructor approval for another approach.

### Problem Overview

This project will simulate disk allocation methods.

The project will allow the user to choose an allocation method from among the three presented in the textbook. It will allow the user to perform disk operations on a simulated disk.

**Details**

The disk will consist of 256 blocks of 512 bytes each. The first block is for the file allocation table. The second block is a bitmap for free space management. The remaining blocks hold data for the files.

Storage for the disk should be kept in an array. The disk should be its own object with a set of public methods to use it. A disk only knows how to read and write blocks by block number.

File names should be allowed up to at least 8 characters. The maximum file size should be 10 blocks. Files are assumed to be at the root level, and subdirectories are not supported.

A user interface should be written to interact with the disk object and to support the menu options shown below. It is the interface that implements most of the processing, the disk is just a storage device.

The simulation should allow the user to choose the allocation method to use when the simulation begins, either through a menu or as a command line argument. For the chained and indexed allocations, a free block should be randomly chosen. For a contiguous allocation, choose the first contiguous set of blocks large enough for the request.

The user interface should let the user do the following things:

1) Display a file

2) Display the file table

3) Display the free space bitmap

4) Display a disk block

5) Copy a file from the simulation to a file on the real system

6) Copy a file from the real system to a file in the simulation

7) Delete a file

8) Exit

The format of the file allocation table should be as shown in the slides, with tabs between columns and with each line ending with a newline character.

You should handle problems such as not having enough space in an appropriate way.

**Sample Output**

>project3 contiguous

1) Display a file

2) Display the file table

3) Display the free space bitmap

4) Display a disk block

5) Copy a file from the simulation to a file on the real system

6) Copy a file from the real system to a file in the simulation

7) Delete a file

8) Exit

Choice: 6

Copy from: abc.txt

Copy to: abc.txt

File abc.txt copied

1) Display a file

2) Display the file table

3) Display the free space bitmap

4) Display a disk block

5) Copy a file from the simulation to a file on the real system

6) Copy a file from the real system to a file in the simulation

7) Delete a file

8) Exit

Choice: 2

abc.txt 2 3

1) Display a file

2) Display the file table

3) Display the free space bitmap

4) Display a disk block

5) Copy a file from the simulation to a file on the real system

6) Copy a file from the real system to a file in the simulation

7) Delete a file

8) Exit

Choice: 3

11111000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

## III. Project Guidelines

### Submitting

Submit your project on eLearning. Include in your submission the following files:

1. readme.txt. A readme file describing how to compile and run your project
2. summary.doc A Word document for the summary
3. Your source files

### Academic Honesty

All work must be your own. If cheating is suspected, you will be referred to the Judicial Affairs Office for further discussion. Copying may be detected in a number of ways, including by software which compares your code with all other students’ source code, by comparison with code on the Internet, or by a visual inspection of your source code.

### Resources

The slides and textbook describe the three allocation methods to be used.