

Drive Sync: A Real-Time Disk Drive Backup Utility

Michael Nicholson

Outline

- **Program Flow**
- **Program Components**
 - GUI
 - GUI Action Classes
 - Sync Server
 - Incoming Sync
 - Outgoing Sync
- **Demonstration with Debug Output**

Program Flow

- **Program Flow**
- Instantiate actions and put them into a hash table
- Initialize vectors to hold incoming and outgoing syncs
- Instantiate SyncServer to begin listening to incoming connections
- Instantiate the GUI and set visible
- At this point the program is waiting for user actions and listening for incoming connections – both of which can happen at the same time by utilizing multiple threads

GUI Action Classes

- These classes correspond to the 10 actions the user is capable of sending to the program through the GUI's buttons
- Each class contains its own actionPerformed method and is solely responsible for performing the desired action
- a_AcceptNewSync
- a_BrowseDest
- a_BrowseOrig
- a_CancelNewSync
- a_DailyTime
- a_Exit
- a_NewSyncDialogue
- a_RealTime
- a_RemoveSync
- a_SelectMaxBandwidth

SyncServer

- Handles incoming connections issued by an instance of this program running from another computer on the local network
- When a new incoming connection is detected the necessary parameters needed to maintain this sync are collected and used to create an instance of IncomingSync

```
while(true) {  
    connection = server.accept();           // blocking  
    destinationDirectory = getDestinationDirectory();  
    numFiles = getNumberOfFiles();  
    filenames = getFileNames();  
    new IncomingSync(destinationDirectory, numFiles, filenames);  
}
```

Incoming Sync

- Responsible for maintaining the new sync created by SyncServer
- Periodically checks for new data that has been sent
- When new data is detected it is written to the specified directory

```
while(true) {  
    while(!destinationFiles.isEmpty()) {  
        outputFile = destinationFiles.removeElementAt(0);  
        write(outputFile);  
    }  
    if(newFilesToTransfer())  
        destinationFiles = getNewFiles();  
}
```

Outgoing Sync

- Responsible for maintaining the new sync created by a `_AcceptNewSync`
- Handles syncs that are initiated from **this** instance of the program

```
while(true) {
    if(connection == null)
        connection = makeNewConnection();
    if(newFilesToSend()) {
        connection.write(destinationDirectory);
        connection.write(numFiles);
        connection.write(filenamees);

        while(!files.isEmpty()) {
            currentFile = files.removeElementAt(0);
            connection.write(currentFile);
        }
    } else
        sleep(5000);           // wait 5 seconds
}
```