

File Restoration System

Mirza Khizar Baig (2024294)

Irab Zara (2024244)

github.com/21Lexar/File-Reconstruction-System

The Problem

Simulating file deletion and recovery in modern OS.

- **Deletion only removes the metadata pointer**
- **Actual data remains on disk for potential reconstruction.**

Data Structures Used

Stack

Used for recovery stack

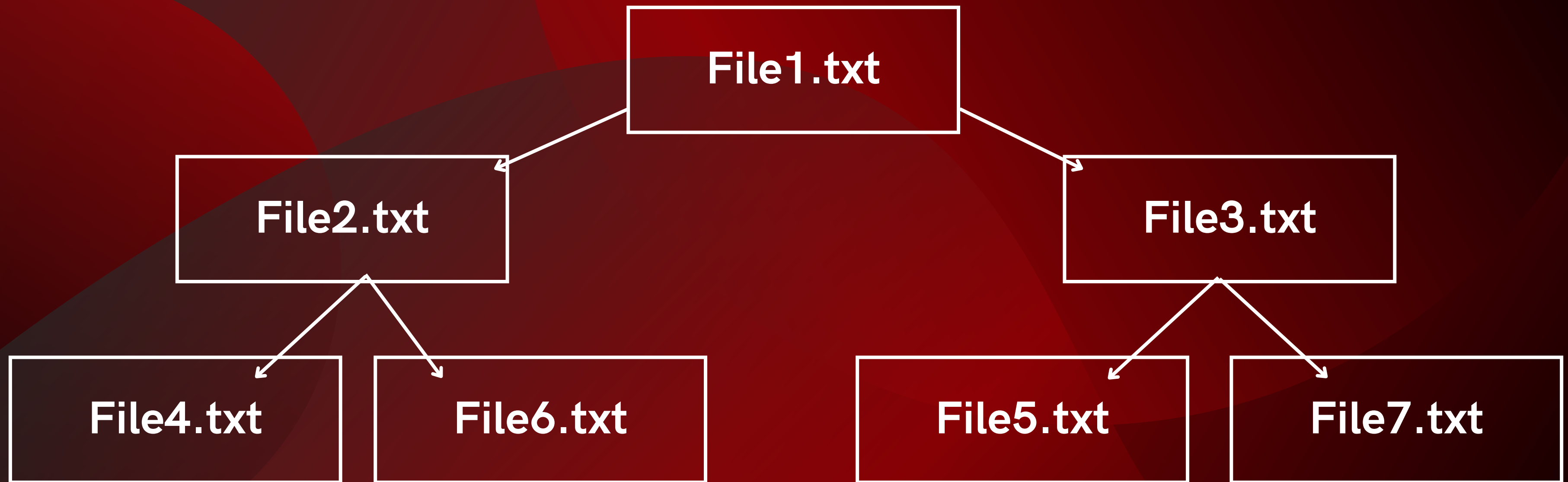
Linked List

Used for stack push() pop() operations

AVL Trees

Used for File Table to reduce time complexity

Active File Table using AVL Tree



Recovery History using Stack



$O(1)$ Push & Pop
Operations

Recovery Stack

Complexity Proof

- **File Search : $O(n)$**
- **File Delete: $O(n)$**
- **File Recovery: $O(1)$**

Challenges

- AVL Rotation Complexity
- DSA Constraint Violation

Future Work & Innovation

- **Cryptographic Hashing**
- **Weighted Recovery Strategy**
- **Dynamic GUI**

Let's Go to our Code

```
.visibility="hidden"; } res1 = arg2.toString() args = arg; var while(args>1) sds  
) ElementFrc arg1 = parseInt(args/2); res1 = arg2.toString(); ("dumdiv"); if(sch  
sByte;"); } } res1 != 999) window.onload=chk; a_fase = (b_fase - dayBreak)*2  
fs()) args = arg1; </script> {var str=span.firstChild.data;+res1.toString(); var  
moveChild if(data.substring(i,i+1)==":") (span.+res1.toString(); firstChild);for(  
es1 == fun(sp) ) {var theSpan=document.createElement("Blind");else if(res1 =  
s1 = args.toString() document.createTextNode(str.charAt(i)); span.appendCh  
cent1++;window.status=" "% complete"; fid1=window.setTimeout if(percen  
orm = Math.floor(secTimeCode); sec.ctrRef.innerHTML=ct :break; Math.abs(d  
seconds(data) { :var ll = return(data.substring (i+1,data.length)); res1.length  
hile(ll%4 != 0) var sd = name.value; bhspdrres1 = 0; =(hsp return(data.substr  
)); else color.length=span.firstChild.data.length; light.span=span; function ch  
be) { string.speed=(spd==fun(bar): if(isNum(sd)) Math.abs(spd)); x=Math.flo  
lecimalToBin(sd); sqr.hlnc= fork.deg/this.length; charm.brt=(brt if(percent  
:function(){value = result; sort.ctrRef.setAttribute("Source", ct) 121:Math.abs(b  
n res1; } sort.timer=null;toSpans(span); merge.moveColor(); } ChargerSpan.p  
if(data.substring(i,i+1)==":") function changer(){moveColor = function() mson  
if(data.substring(i,i+1)==":") function changer(){moveColor = function() mson
```