CS502 Project2 Documentation

1. What is included.

|  |  |  |
| --- | --- | --- |
| Type | Content | PS |
| File System | SVC: FORMAT |  |
| SVC: OPEN\_DIR |  |
| SVC: CREATE\_DIR |  |
| SVC: CREATE\_FILE |  |
| SVC: OPEN\_FILE |  |
| SVC :CLOSE\_FILE |  |
| SVC :WRITE\_FILE |  |
| SVC:READ\_FILE |  |
|  | SVC: DIR\_CONTENTS |  |
| Memory Management | Faulthandler Design |  |
| Swaparea Design |  |
| LRU Alg Implement |  |

1. High Level Design
2. File System

Control

status

Disk

Directory

File

PCB

SVC

Status

Instructions

Process Control Block

Other

SVC

FILE READ&WRITE

Control

Information

Data

Write

Write

Read

Read

DiskX8

User

Control

1. Memory Management

Y

N

Judge the victim is in the swaparea

Put victim to the swaparea

Allocate that victim page to VPN

Allocate that victim page to VPN

Allocate that victim page to VPN

Put victim to the swaparea

Allocate that victim page to VPN

Y

N

Y

Find the physical page and allocate that page to VPN

Judge the victim is in the swaparea

LRU Algorithm to find victim

LRU Algorithm to find victim

N

N

Judge if there’s physical page available

Y

VPN

Judge pid in the swaparea

PID

1. Justification
2. For the file management part, I give put the header and sector location not just in the disk but also in the pcb which I can do a lot of things in a very simple way like finding the father directory without reading back data from the disk, which saves a lot of time. Also, when we write or read a file, we need to know the locations of its sectors. We can easily find the sectors in this way. And the sector I allocate is one by one, first come first serve. Because we have no delete file svc, we don’t need to care the piece of sectors is little or “fragment”. So FCFO is very efficient in this system.
3. For the memory management, I set the area in each disk 1024 sectors which correspond every vpn in the process. And I notice we only deal with most 5 process, so 8 disks are enough to handle this workload. In this way, we do not need another algorithm to decide and find the sector contains the data we need. Also, I cache a global physical page with their relating vpn and process ID, which we can refer in a very direct way.
4. Unique Part

In the file management, I cache an array of information for each node. With this array, we can easily find the connection between each node. Like for the DIR\_CONTENT svc, we can just find the nodes whose father directory is the one we want and return the information we find in the nodes.

For calculating the size of file, I calculate the size only when the svc CLOSE\_FILE is called, so it saves a lot of time in disk writing.

1. Bug

Though I try to design the multiprocessor feature, but it comes with a lot of problems, and I don’t know where to fix those problems. Like the dispatcher will call the terminated process to run again, the first process will never be suspended.

Another problem is lock, I was stuck by a lock code “READ\_MODIFY(MEMORY\_INTERLOCK\_BASE + 24, DO\_LOCK, SUSPEND\_UNTIL\_LOCKED,&LockResult);” one day, though I come through it by deleting it, but it still confuses me why it behaves like that. Maybe I was in a deadlock, and I didn’t notice that. Or it could be something else.