CS 502
Project 2 Report

Jiaming Nie 369577520

Content

List of Tables

List of Figures

1 Architectural Document

1.1 Design List

In CS502 project 2, the functions that are implemented could be classified as the following:

- The operation of the Memory management which allows the process to access the physical frame using the Page table of the process.
- The dispatcher of the limited physical frame with the implementation of simple FIFO (First In First Out) algorithm based on the Queue operations.
- The associated functions that serve with the resource dispatcher of the physical frame.

In the operating system Z502, there are 1024 Virtual pages and 64 Physical frames.

1.1.1 Functions in MemoryManagement.h

This header file contains the declarations of the functions for the implementation of the FIFO algorithms. The content is illustrated in table 1.

Table 1: Functions in MemoryManagement.h

Function Name	Description
*initialLRUQueue	The information of the memory addresses which are using the physical frame.
EnQueueNewFrame(LRUQueue)	Enqueue new memory addresses into the LRU queue.
DeQueueLeastUsedFrame(Queue)	Remove the first memory address coming in the queue, identified as the victim.
IsFreeFrameExist(Queue)	Examine whether there are free frames available.
GetFreeFrame(Queue)	Get the free frames if they are available.

1.1.2 Memory Address Structure

The structure of memory information is defined on the following table 2.

Table 2: MemoryInfo Struct

Variables	Description
int PhysicalFrame	The physical frames the memory address is using.
int VirtualPageNumber	The virtual page the memory address is using.
long PageTable	The page table the memory address is using. It's the same as its process.
int DiskID	The disk id which it will write its data.
int ProcessID	The process which is using the memory address.

1.1.3 Related Functions that Used

In the access of the physical frames and the input and output of the data, some related functions are also used. In the resources scheduling, once the victim is chosen, its data will be written to the disks.

Table 3: Related Functions

Z502ReadPhysicalMemory()	Built in the z502.c which can read the data from the memory without the use of page table.
Z502WritePhysicalMemory()	Built in the z502.c which can write the data to the memory without the use of page table.
Physical_Disk_Read()	Implemented in project 1. Read the data from the disks.
Physical_Disk_Write()	Implemented in project 1. Write the data to the disks.

1.2 High Level Design

The flowchart of the mechanism when process access the physical memory using page table. It demonstrates how the page fault (Type: Invalid memory) is invoked and how the system handles the page fault.

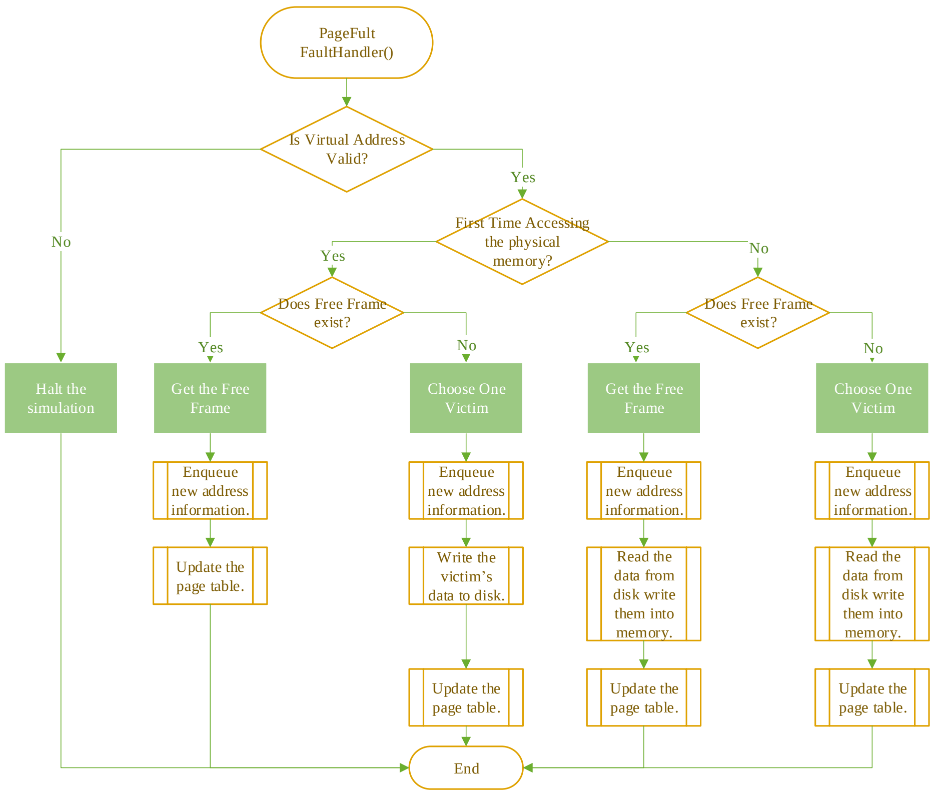
PageFult FaultHandler()
Is Virtual Address Valid?
No
Yes
First Time Accessing the physical memory?
Yes
No
Does Free Frame exist?
Yes
No
Does Free Frame exist?
Yes
No
Halt the simulation
Get the Free Frame
Choose One Victim
Get the Free Frame
Choose One Victim
Enqueue new address information.
Enqueue new address information.
Enqueue new address information.
Enqueue new address information.
Update the page table.
Write the victim's data to disk.
Read the data from disk write them into memory.
Read the data from disk write them into memory.
Update the page table.
Update the page table.
Update the page table.
End

Figure 1: Flow chart of the System Handles the Page Fault

1.3 Justification of High Level Design

1.3.1 Address Legality

The first step when the page fault is invoked is to check whether the memory address is illegal.

The total page table number is 1024 if the the memory address is using exceeds this limit, the address is illegal.

1.3.2 Repeated Accessing

When the page fault is invoked, the memory address may not be used for the first time. If the process used the address for the first time, which means that the system has no record of the I/O data of this address. There are two occasions that the system will handle:

1. If the address accesses the memory for the first time, the Z502 system will need to record its I/O data, which is to write them into the disk for future use.
2. If it is not the first time for the address accessing the memory, the Z502 system will need to read the data from the disk and write them into the memory.

1.3.3 Resources Scheduling

The physical frames are limited, when processes access the memory there are no available physical frames for most time.

One simple FIFO algorithm is adopted for the resources scheduling. Its description is on the following:

- When there is available physical frames, then use the free frame.
- If there is no free frames, pick one victim from the queue where storing the memory addresses which are accessing the physical frames.
- The algorithm is FIFO, in the process of picking the victim, choose the victim with longest stay in the queue, or in the physical memory.

1.4 Unique Features of the Project

1.4.1 Test 26 Seizure

The project is compiled by Visual studio 2017 and GCC. In the debug mode of Visual studio 2017, the system performs well while under the compilation, the system will sometimes be stuck.

The reason that causes the seizure could be the problems arising in the read data from disk according to the infer.

1.4.2 Test 10 Different Behaviors

Test 6 will create the test 10 as its child process, the result of the checking disk will be different on different debug modes of Visual Studio 2017 and GCC.

1.5 Anomalies and Bugs

The LRU (Least Recently Used) algorithm, it cannot be implemented by using Fault Handler only.

In the theory case, when the new page fault is invoked, the fault handler will add new item to the queue. When picking the victim, the queue will remove the first element.

When one memory address is accessing the physical memory, its virtual page number is valid. When the memory address accesses the physical address for the second time, in the queue, the position of the memory address should be the last one, which means it's not recently used.

All the page fault and queue operations are performed in the fault handler, if the fault handler is not invoked, then the algorithm cannot could not perform as theory case.