COP4610: Project 1 – Memory Location Finder

Overview

The Memory Management module in an operating system needs to perform several calculations in order to serve and assist the CPU. The CPU generates and forwards merely the logical addresses and the Memory Manager must figure out the physical address locations for these logical addresses. In this project, we will try to write a high-level language program that will perform calculations for a very small group of the memory manager functions. The idea behind this project is to enable a better understanding of paging-based memory allocation for the students.

Learning Objectives

The focus of this assignment is on the following learning objectives:

- Be able to calculate the frame requirement for a process based on the agreed page/frame size when the process informs the memory manager about the overall process size
- Be able to identify the location of an address in the logical address space
- Be able to map and report the absolute location in physical memory for a logical address
- Be able to calculate and port the offset value used for the generation of the physical address which is to be passed through the address protection hardware

Prerequisites

- To complete this project, you need to make sure that you have the following:
- C/C++ compiler
- A C/C++ IDE or text editor (multiple editors exist for developers)
- An understanding of the material presented in class
- An understanding of the material covered in the textbook.

Problem Description

You are to implement a program that would solve questions like the one below:

A program that spans 1540 locations has just become a process.

The frame size in the memory is 350 locations per page.

The Page Map table (PMT) for the process is ... - program will allow the user to enter frame numbers

What will be the physical memory location for the 770th logical memory location (somewhere in the middle of the process' pages) in terms of the frame number and offset for the location?

From the start of the memory map, counting the first location as 1,

What location number by simple count is it going to be in the physical memory?

The list of requirements are as follows:

General requirements for the program:

1. The program accepts the size of the process in logical number of locations and the page/frame size agreed – as user inputs.

2. Based on the user inputs as above, the program calculates the size of the page map table. It then asks the user for the frame numbers that should be assigned to the respective logical page numbers starting from logical page number 0. After it gets the input, it prints the page map table.
3. The program then asks for a logical memory location to be entered by the user for which it reports the page number, the frame number that the page is in, the offset of the location and its absolute location in the memory from the start of the memory map.
4. The program should clear any dynamically allocated memory used and not leave any memory leaks before exiting.

Grading Breakdown

- **[7 pts]** Reporting the number of pages correctly. Care should be taken to calculate the number of pages correctly irrespective of whether the process perfectly spans an integral number of pages or partly uses an additional page.
- **[15 pts]** Accepting the required frame numbers from the user. Care should be taken to accept only as many frame numbers as the legit page numbers for the process.
- **[2 pts]** Printing the Page Map Table.
- **[8 pts]** Finding out the correct page number for the logical memory location in question and reporting it.
- **[2 pts]** Reporting the frame for the logical memory location in question correctly.
- **[8 pts]** Calculating and reporting the offset for the logical memory location in question correctly.
- **[8 pts]** Calculating and reporting the absolute physical location (from the start of the memory map) for the logical memory location in question correctly.

Submission
Points will be deducted for not following these instructions.

Before submitting this project in eLearning make sure that you follow the following steps:

1. Make sure that your name appears at the top of each file. This should be done as a comment for any source code files.
2. If your project spans source code multiple files (.h & .cpp or .c), add all files into a .zip file with the file name “lastname_firstname_project01.zip”.
3. Turn your **zipped up** project into eLearning/Canvas.

SAMPLE RUN 1 (Values in **teal** indicate user input)

```
Enter the number of locations that the program spans, followed by the page/frame size:
1540 350
Number of pages: 5         Page numbers: 0 - 4
Enter the frame numbers for the PMT array:
9
5
11
7
6
The PMT is:

   Page#      Frame#
--------------------------
     0          9
     1          5
```

```
     2          11
     3          7
     4          6
Enter the logical memory location in question: 770

770th logical memory will be on page                = 770/page size
                                                  = 770/350    = 2.2
                                                  = 3(bump up to the higher integer)
Counting from page# 0, the 3nd/st/th page will be page 2 (#0, 1, 2)
Number of locations until before the last page       = 350 x 2     = 700 locations
Number of locations in the last page                 = 770 - 700   = 70 locations
Starting from an offset of 0, the 70nd/st/th location will be at an offset of 69.
--------------------------------------------------------------------------------
From the start of the memory map:
This page is in Frame 11, which is the 12nd/st/th by count as we start from frame# 0
Number of locations covered from start of the map in the first 11 frames  = 11 x 350 =
3850 locations
The last 70 locations are in frame# 11. Total distance from start        = 3850 + 70 =
3920 locations.
So, the 770th location by the logical memory is 3920nd/st/th location in the physical
memory, as per the given PMT.
```

SAMPLE RUN 2

```
Enter the number of locations that the program spans, followed by the page/frame size:
1540 350
Number of pages: 5         Page numbers: 0 - 4
Enter the frame numbers for the PMT array:
9
5
11
7
6
The PMT is:

   Page#      Frame#
--------------------------
     0          9
     1          5
     2          11
     3          7
     4          6
Enter the logical memory location in question: 700

700th logical memory will be on page     = 700/page size
                                    = 700/350 = 2
                                    = 2(bump up to the higher integer)
Counting from page# 0, the 2th page will be page 1 (#0, 1)
Number of locations until before the last page = 350 x 1 = 350 locations
Number of locations in the last page = 700 - 350 = 350 locations
Starting from an offset of 0, the 350th location will be at an offset of 349.
--------------------------------------------------------------------------------
From the start of the memory map:
This page is in Frame 5, which is the 6th by count as we start from frame# 0
Number of locations covered from start of the map in the first 5 frames = 5 x 350 = 1750
locations
The last 350 locations are in frame# 5. Total distance from start = 1750 + 350 = 2100
locations.
So, the 700th location by the logical memory is 2100th location in the physical memory,
as per the given PMT.
```