

## CSCI 3120 Operating Systems

### Assignment 4: Logical/Virtual Address Translation

**Due: 16:00, Dec. 4, 2019**

- **Teaching Assistants:**
  - o Lauchlan Toal (lc790935@dal.ca),
  - o Patricia Kibenge-MacLeaod (p.kibenge@dal.ca)
  - o Hui Huang (huihuang@dal.ca)
- **Help Hours at CS Learning Center (2<sup>nd</sup> Floor of Goldberg Building):**
  - o Tuesdays: 1pm-3pm, Hui Huang
  - o Thursdays: 2pm-4pm, Patricia Kibenge-MacLeaod
  - o Fridays: 12pm-2pm, Lauchlan Toal

---

#### 1. Assignment Overview

In this assignment, you need to design and implement a C program that translates a series of logic addresses into the corresponding physical addresses.

#### 2. Important Note

There is a zero-tolerance policy on academic offenses such as plagiarism or inappropriate collaboration. By submitting your solution for this assignment, you acknowledge that the code submitted is your own work. You also agree that your code may be submitted to a plagiarism detection software (such as MOSS) that may have servers located outside Canada unless you have notified me otherwise, in writing, before the submission deadline. Any suspected act of plagiarism will be reported to the Faculty's Academic Integrity Officer in accordance with Dalhousie University's regulations regarding Academic Integrity. Please note that:

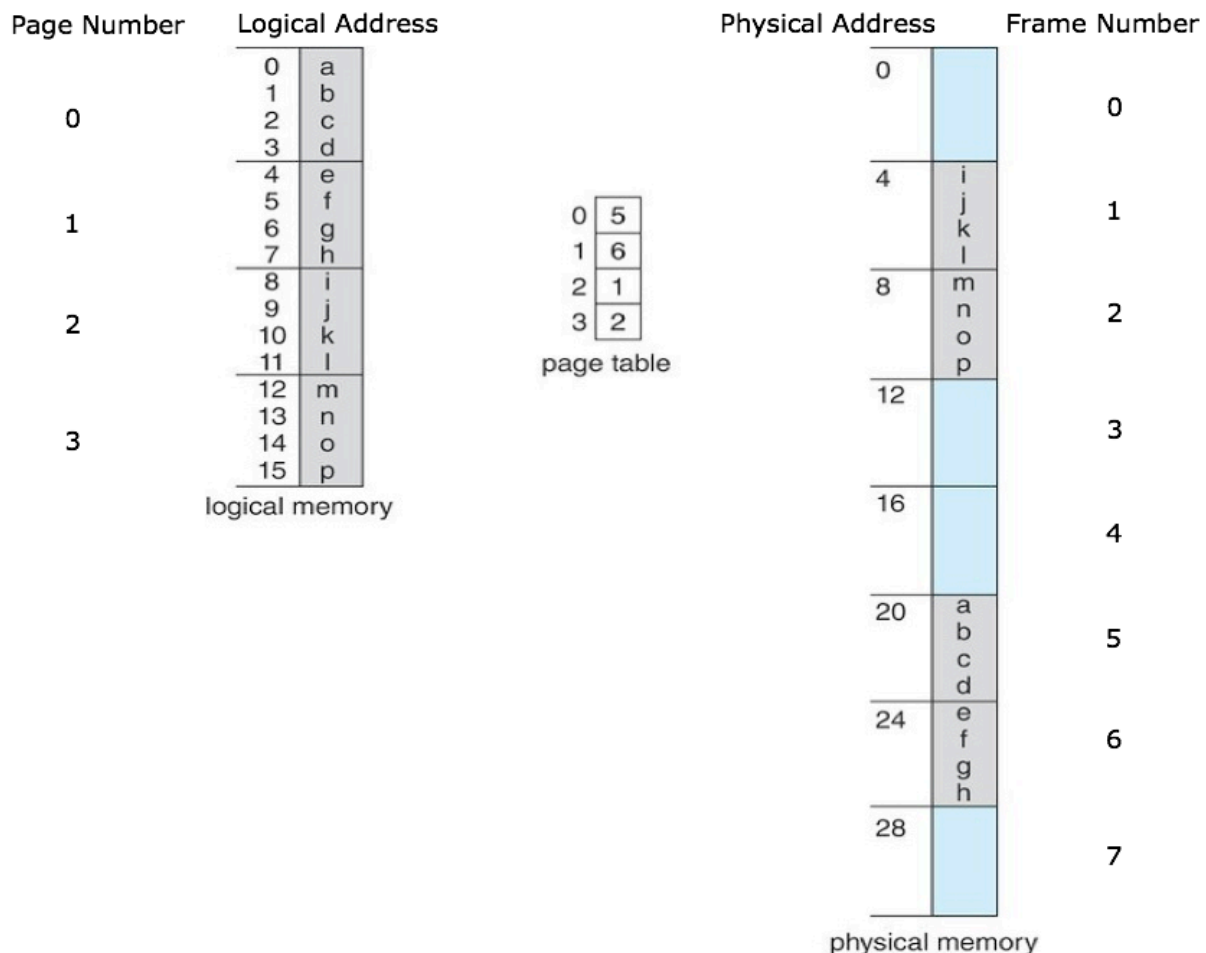
- 1) The assignments are individual assignments. You can discuss the problems with your friends/classmates, but you need to write your program by yourself. There should not be much similarity in terms of coding.
- 2) When you refer to some online resources to complete your program, you need to understand the mechanism and write your own code. In addition, you should cite the sources via comments in your program.

#### 3. Detailed Requirements

1) Overview: Paging is a widely-used memory management scheme. With paging, a logical address (i.e. virtual address) needs to be translated to the corresponding address in order to access the correct memory unit. In this assignment, you need to design and implement a C

program that translates a series of logic addresses into the corresponding physical addresses for paging-based computers.

2) Logical Memory and Physical Memory: For simplicity, we assume that the computer under investigation is byte addressable. Namely, each byte has a unique address. In addition, the size of logical memory is 16 bytes, the size of page/frame is 4 bytes, and the size of physical memory is 32 bytes. Consequently, there are 4 pages and 8 frames. And there are 4 entries in the page table. The following figure shows the details of the logical and physical memory.



3) Input Files: Your program needs to retrieve a series of logical addresses from a file named "LogicalAddress.txt", which is located in the same directory as your program. Each row of "LogicalAddress.txt" includes a logical address to be translated. A sample "LogicalAddress.txt" is provided in this assignment. However, the TA will use different addresses to test your program. Here is the content of the sample "LogicalAddress.txt":

3  
0  
13  
4

In addition, your program needs to retrieve the page table information from a file named "PageTable.txt", which is located in the same directory as your program. Each row of "PageTable.txt" includes two fields, page number and the corresponding frame number. These two fields are separated by a comma. A sample "PageTable.txt" is provided in this assignment. However, the TA will use different page tables to test your program. Here is the content of the sample "PageTable.txt":

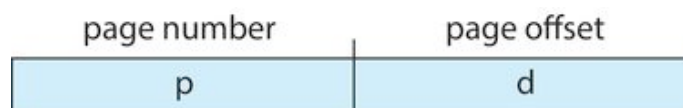
```
0,5
1,6
2,1
3,2
```

4) Output File: After retrieving the information from "LogicalAddress.txt" and "PageTable.txt", your program needs to translate the logical addresses into the corresponding physical addresses, and place the resulting physical addresses in a file named "PhysicalAddress.txt", which is in the same directory as your program. Each row in "PhysicalAddress.txt" should include one physical address for the corresponding logical address. Namely, the first row in "PhysicalAddress.txt" should include the physical address corresponding to the logical address in the first row of "LogicalAddress.txt", etc. Here is the content of the "PhysicalAddresses.txt" for the sample "LogicalAddress.txt" and "PageTable.txt" (note that the lecture notes include the detailed analysis for this example):

```
23
20
9
24
```

5) Address Translation Procedure: To translate a logical address that is composed of page number field and page offset field, the following three steps need to be executed.

- a) Step 1: Extract the page number and page offset field.
- b) Step 2: Extract the corresponding frame number from the page table.
- c) Step 3: Replace the page number in the logical address with the frame number.



Note that there are varied methods to extract page number and page offset from a logical address. One simple method is based on bitwise operators in C language. For example, assume that  $A=1110_2$ ,  $B=0011_2$ ,  $C=1100_2$ , and bitwise AND is denoted as "&", then we have:

- a)  $A \& B = 0010_2$ ;
- b)  $A \& C = 1100_2$ .

Note that you do not have to use this field extraction method. As long as you follow the above 3-step procedure to complete address translation, you can use any method to extract page number and page offset.

6) Additional Information:

- a) You can assume that there are at most 10 rows in “LogicalAddress.txt”.
- b) A tutorial on bitwise operators in C language can be found here:  
[https://www.tutorialspoint.com/cprogramming/c\\_bitwise\\_operators.htm](https://www.tutorialspoint.com/cprogramming/c_bitwise_operators.htm)
- c) If you choose to use bitwise operators in your program, it is better to use “unsigned int” as the data type for logical and physical addresses.
- d) Compiling and running your program on bluenose.cs.dal.ca should not lead to errors or warnings.
  - a. To compile and run your program on bluenose, you need to be able to upload a file to or download a file from bluenose. Here is a tutorial on bluenose downloading/uploading: <https://web.cs.dal.ca/~society/#/>

7) Readme File: You need to complete a readme file named “Readme.txt”, which includes the instructions that the TA could use to compile and execute your program on bluenose.

8) Submission: Please pay attention to the following submission requirements:

- a) You should place “Readme.txt” in the directory where your program files are located.
- b) Your program files and “Readme.txt” should be compressed into a zip file named “YourFirstName-YourLastName-ASN4.zip”. For example, my zip file should be called “Qiang-Ye-ASN4.zip”.
- c) Finally, you need to submit your zip file for this assignment via brightspace.

Note that there is an appendix at the end of this document, which includes the commands that you can use to compress your files on bluenose.

## 4. Grading Criteria

The TA will use your submitted zip file to evaluate your assignment. The full grade is 20 points. The details of the grading criteria are presented as follows.

- “Readme.txt” with the correct compilation/execution instructions is provided [1 Point]
- Logical addresses are retrieved from “LogicalAddress.txt”. [2 Points]
- Page table information is retrieved from “PageTable.txt”. [2 Points]
- The output is placed in “PhysicalAddresses.txt”, which is in the same directory as your program. [2 Points]
- Logical addresses are correctly translated into physical addresses using the 3-step procedure summarized in Section 3.5:
  - Step 1: Extract the page number and page offset field. [4 Points]
  - Step 2: Extract the corresponding frame number from the page table. [4 Points]
  - Step 3: Replace the page number in the logical address with the frame number. [4 Points]
- Proper programming format/style (e.g. release the memory that is dynamically allocated when it is not needed any more; proper comments; proper indentation/variable names, etc.). [1 Point]

Please note that when “Readme.txt” is not provided or “Readme.txt” does not include the compilation/execution instructions, your submission will be compiled using the standard command `gcc -o A4 A4.c` (or your filename), and [you will receive a zero grade if your program cannot be successfully compiled on bluenose.](#)

## 5. Academic Integrity

At Dalhousie University, we respect the values of academic integrity: honesty, trust, fairness, responsibility and respect. As a student, adherence to the values of academic integrity and related policies is a requirement of being part of the academic community at Dalhousie University.

### *1) What does academic integrity mean?*

Academic integrity means being honest in the fulfillment of your academic responsibilities thus establishing mutual trust. Fairness is essential to the interactions of the academic community and is achieved through respect for the opinions and ideas of others. Violations of intellectual honesty are offensive to the entire academic community, not just to the individual faculty member and students in whose class an offence occur (See Intellectual Honesty section of University Calendar).

### *2) How can you achieve academic integrity?*

- Make sure you understand Dalhousie’s policies on academic integrity.
- Give appropriate credit to the sources used in your assignment such as written or oral work, computer codes/programs, artistic or architectural works, scientific projects, performances, web page designs, graphical representations, diagrams, videos, and images. Use RefWorks to keep track of your research and edit and format bibliographies in the citation style required by the instructor. (See <http://www.library.dal.ca/How/RefWorks>)
- Do not download the work of another from the Internet and submit it as your own.
- Do not submit work that has been completed through collaboration or previously submitted for another assignment without permission from your instructor.
- Do not write an examination or test for someone else.
- Do not falsify data or lab results.

These examples should be considered only as a guide and not an exhaustive list.

### *3) What will happen if an allegation of an academic offence is made against you?*

I am required to report a suspected offence. The full process is outlined in the Discipline flow chart, which can be found at:

<http://academicintegrity.dal.ca/Files/AcademicDisciplineProcess.pdf> and includes the following:

- a. Each Faculty has an Academic Integrity Officer (AIO) who receives allegations from instructors.

- b. The AIO decides whether to proceed with the allegation and you will be notified of the process.
- c. If the case proceeds, you will receive an INC (incomplete) grade until the matter is resolved.
- d. If you are found guilty of an academic offence, a penalty will be assigned ranging from a warning to a suspension or expulsion from the University and can include a notation on your transcript, failure of the assignment or failure of the course. All penalties are academic in nature.

*4) Where can you turn for help?*

- If you are ever unsure about ANYTHING, contact myself.
- The Academic Integrity website (<http://academicintegrity.dal.ca>) has links to policies, definitions, online tutorials, tips on citing and paraphrasing.
- The Writing Center provides assistance with proofreading, writing styles, citations.
- Dalhousie Libraries have workshops, online tutorials, citation guides, Assignment Calculator, RefWorks, etc.
- The Dalhousie Student Advocacy Service assists students with academic appeals and student discipline procedures.
- The Senate Office provides links to a list of Academic Integrity Officers, discipline flow chart, and Senate Discipline Committee.

## **Appendix: How to Use Zip and Unzip on Bluenose**

To compress:

```
zip squash.zip file1 file2 file3
```

To uncompress:

```
unzip squash.zip
```

this unzips it in your current working directory.