**Project 2: Pager – A Virtual Memory Manager**

Kyungchan Im

Grand Canyon University

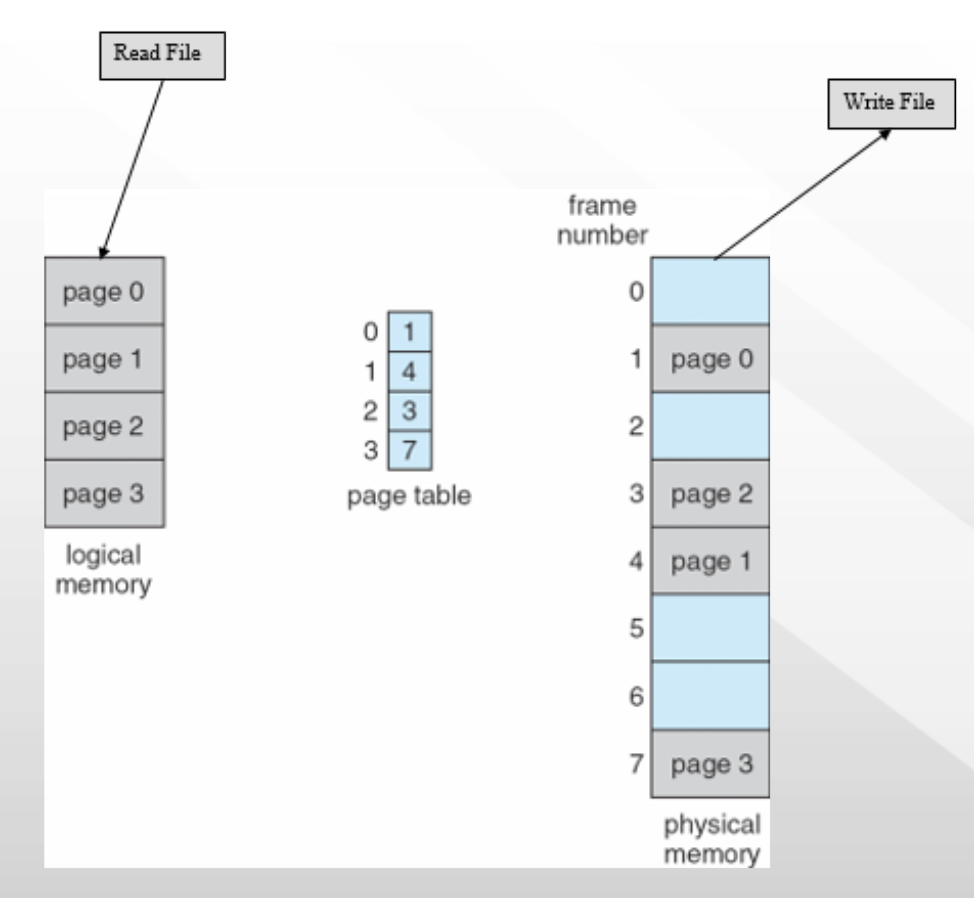
Course number: CST - 315

Professor Ricardo Citro

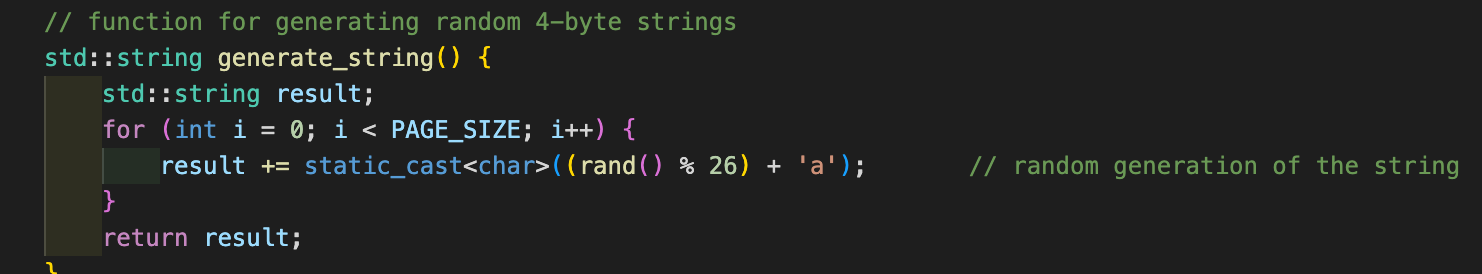
Due Mar 26, 2023

**Project 2: Pager – A Virtual Memory Manager**

This is the project to demonstrate the Page and Virtual Memory Management Process. The program will generate four bytes of strings and store in a text file. Once the file is generated, it will read the strings within a text file and store in a particular page in a logical address define by the developer with the maximum number of 4 areas. It will allocate each individual strings with page numbers (from 0 to 3) and will assign random physical address number (from 0 to 7).

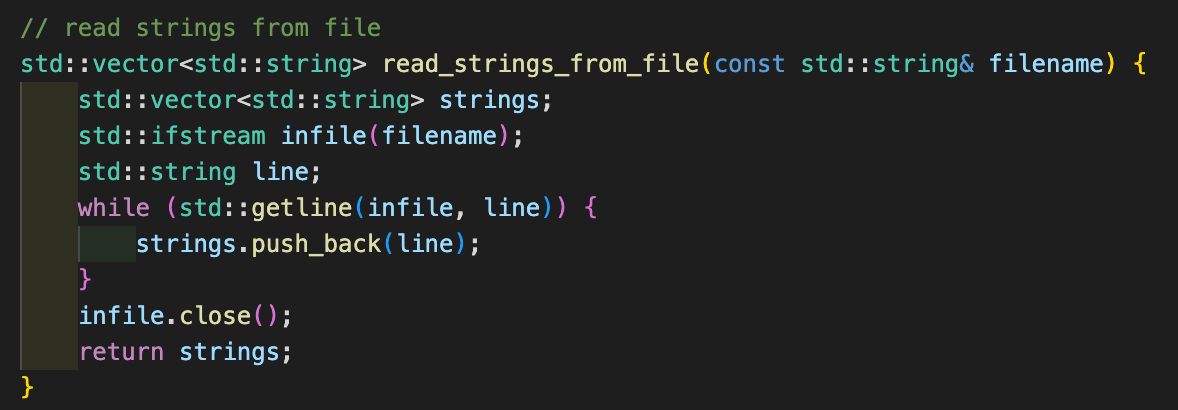


The program generates random 4-byte strings using the generate\_string function and writes them to a text file called "strings.txt" using the write\_strings\_to\_file function.

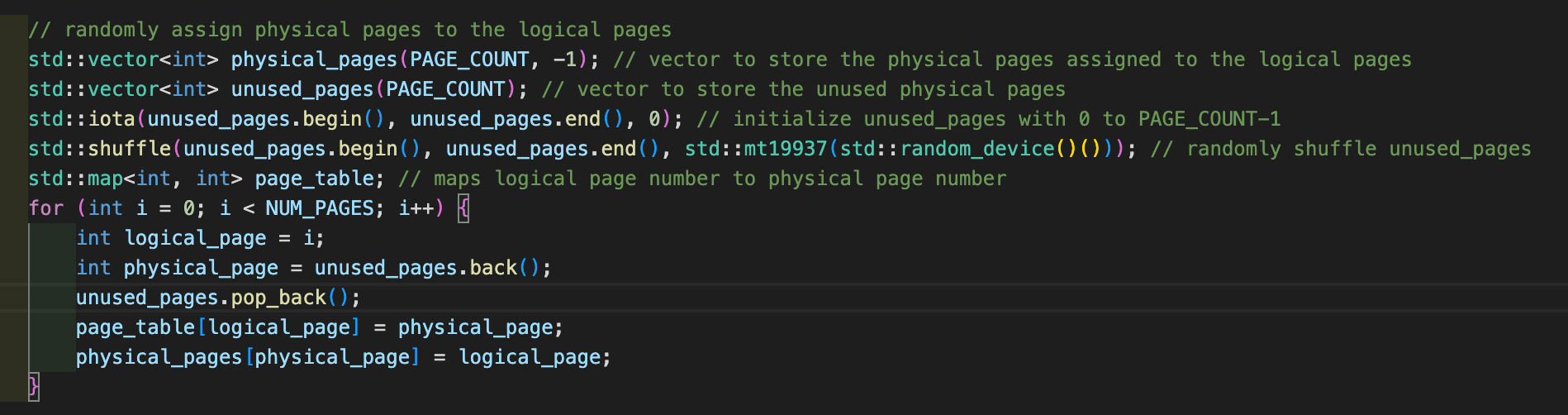
Text

Description automatically generated

The program then reads the strings from the text file using the read\_strings\_from\_file function.



Next, the program randomly assigns physical pages to the logical pages using the std::shuffle function, and stores the mapping between logical pages and physical pages in a std::map. Finally, the program prints out the page numbers and physical page numbers to a file called "output.txt", using the mapping stored in the std::map.



The "output.txt" file contains the page numbers and corresponding physical page numbers for the randomly generated 4-byte strings. Each line of the file shows the logical page number, the string that was generated, and the physical page number that was randomly assigned to that logical page. The file is generated by the program and contains one line for each string, with the information separated by "->" symbols.

Graphical user interface, text

Description automatically generated

[ Execution ]

The program generates the strings.txt file first, and it will read the strings from strings.txt. Once it is added to the page table, it will assign physical frame address from 0 to 7 and finally it will print out in the format of page number, strings, and physical address.

Graphical user interface, text, application

Description automatically generated