**Programming Assignment Three**

Write a program to simulate a process of translation from a logical address to physical address.

**Assumptions**

1. Assume the file ***la.txt*** includes the sequence of generated addresses from CPU.
2. Use a part of memory as backing store that store data for a process.
3. The backing store size is 128 bytes
4. The size of process ***p*** is 128 bytes.
5. The contents of ***p*** is included in a file ***pdata.bin*** which is a binary file.
6. Use a part of memory as RAM. The size of physical memory is 256 bytes, from 0 to 255. All the physical memory is available, allocating starting from beginning in sequence. That is, allocate frame 0 first, then frame 1, then frame 2….
7. The size of a frame is 32 bytes, i.e., 5 bits for the offset in a frame, total number of frames is 8.
8. At beginning, no page table is available for process ***p***.

**Requirements**

Write a program to

1. Setup a simulating backing store in memory. Read the data from ***pdata.bin*** to this backing store.
2. Initialize a page table for process ***p***, set the frame number to be -1 for each page, indicating that the page is not loaded into memory yet.
3. Read logical addresses one by one from ***la.txt*.**
4. For each logical address,
   1. if its page has been loaded into physical memory, simply find the frame number in the page table, then generate physical address, find and print out the physical address and data inside this address.
   2. if the page is used for the first time, i.e., in page table, its frame number is -1，then the page that contains this address should be loaded into a free frame in physical memory (RAM). Then update the page table by adding the frame number to the right index in the page table. Then repeat 4a).

Refer to Figure 1 for the relationships and how physical memory, backing store, and CPU are simulated.

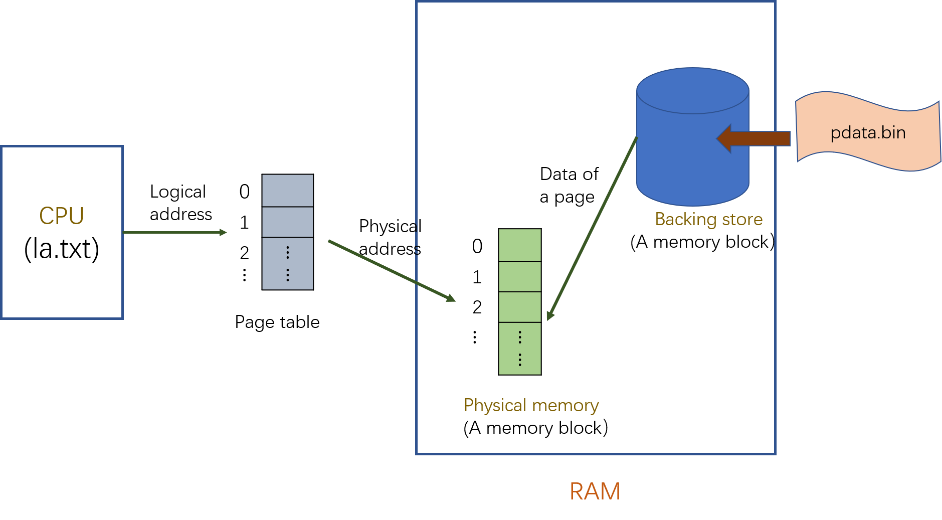


Figure 1 How physical memory, backing store and CPU are simulated in this program assignment

**Hints:**

1. use a memory block pointed by a pointer or use an array as a simulation of backing store
2. use functions *fread* or *mmap* for the binary file read. Search through the Internet for the usage of these functions.
3. Use an array to simulate the memory.
4. Use bit operators &, **|**, <<, and >> to get the bits in a logic address or form a physical address
5. Use **char** for the type of data in the process, use **unsigned char (8 bits)** for the type of address.

**Coding & Submission**

1. Using pure C to finish this program.
2. Put all the codes in one .c file named **PA3\_#####.c**, replace “#####” as the **last 5 digits** of your student ID.
3. Put pdata.txt and la.txt in the same folder as PA3\_#####.c, which the need .txt file can be **open directly by filename** instead of absolute path.
4. Submit **only** the .c file mentioned above.

**Sample Output (next page)**

