Part 1: Memory Management – Paging

Team Member

shengbo lou: 28530995
zhongce ji: 28551884
yucheng lu: 28411887

Compile

Go to the lab03/part1 folder

- compile: `make`
- run: `./part1`

Design

We use the C language to complete the first part of the lab3.
In order to get the offset and page from the virtual address, we will use mask, mask will help us extract the page number and offset number from the virtual address.

- part1.c
  This file have a class which is main class. In the main, first we will read first two lines from the input file. Here we will use fscanf to read each lines

```
fscanf(fp,"%i",&offset_num);
fscanf(fp,"%i",&page_num);
```

For the first line, we will read the offset number from the input file and assign it to the unsigned int offset_num. For the second line, we will read the page number from the input file and assign it to the unsigned int page_num. After that we will will create the mask for both offset number and page number:

```
unsigned int offset_mask = (1<<offset_num)-1;
unsigned int page_mask = ((1<<page_num)-1)<<offset_num;
```

For the first line we move 1 to left for offset number then we minus 1 to get the mask of the offset. Because after that we will have 1 for the position of the offset number. For the second line we move 1 to left for page number, then minus to get the mask of the page. Then we have to move the mask to left for offset number. Then the mask will be the right position which is the page number's position.
Then we jump into the while loop, to read each virtual address from the input:

```
unsigned int offset = tmp & offset_mask;
unsigned int page = tmp & page_mask;
```

For the first line, we get the offset of that virtual address by using & operation.
For the second line, we use the same operation to get the page number. After that, we print out the output for each virtual address.