CS-304 Assignment 3

LINEAR PAGE TABLE

In this homework, you will use a simple program, which is known as `paging-linear-translate.py`, to see if you understand how simple virtual-to-physical address translation works with linear page tables. See the README for details.

Questions

1. Before doing any translations, let's use the simulator to study how linear page tables change size given different parameters. Compute the size of linear page tables as different parameters change. Some suggested inputs are below; by using the `-v flag`, you can see how many page-table entries are filled. First, to understand how linear page table size changes as the address space grows, run with these flags:

   ```
   -P 1k -a 1m -p 512m -v -n 0
   -P 1k -a 2m -p 512m -v -n 0
   -P 1k -a 4m -p 512m -v -n 0
   ```

   Then, to understand how linear page table size changes as page size grows:

   ```
   -P 1k -a 1m -p 512m -v -n 0
   -P 2k -a 1m -p 512m -v -n 0
   -P 4k -a 1m -p 512m -v -n 0
   ```

   Before running any of these, try to think about the expected trends. How should page-table size change as the address space grows? As the page size grows? Why not use big pages in general?

2. Now let's do some translations. Start with some small examples, and change the number of pages that are allocated to the address space with the `-u flag`. For example:

   ```
   -P 1k -a 16k -p 32k -v -u 0
   -P 1k -a 16k -p 32k -v -u 25
   -P 1k -a 16k -p 32k -v -u 50
   -P 1k -a 16k -p 32k -v -u 75
   -P 1k -a 16k -p 32k -v -u 100
   ```

   What happens as you increase the percentage of pages that are allocated in each address space?

3. Now let's try some different random seeds, and some different (and sometimes quite crazy) address-space parameters, for variety:

```
-P 8   -a 32    -p 1024 -v -s 1
-P 8k  -a 32k   -p 1m   -v -s 2
-P 1m  -a 256m  -p 512m -v -s 3
```

Which of these parameter combinations are unrealistic? Why?

4. Use the program to try out some other problems. Can you find the limits of where the program doesn't work anymore? For example, what happens if the address-space size is *bigger* than physical memory?

MULTILEVEL PAGE TABLE

To understand how a multi-level page table works, you will use the program ***paging-multilevel-translate.py;*** see the README for details.

1. With a linear page table, you need a single register to locate the page table, assuming that hardware does the lookup upon a TLB miss. How many registers do you need to locate a two-level page table? A three-level table?

2. Use the simulator to perform translations given random seeds 0, 1, and 2, and check your answers using the `-c` flag. How many memory references are needed to perform each lookup?

3. Given your understanding of how cache memory works, how do you think memory references to the page table will behave in the cache? Will they lead to lots of cache hits (and thus fast accesses?) Or lots of misses (and thus slow accesses)?

Submission Instructions:

1. You must submit a pdf file containing answers to all the questions above in order.

2. Place this **pdf file** along with the **program execution output files** and any other files you wish to submit in your submission directory, with the directory name being your roll number **(18xxxxxxx)** and make two subdirectories one for, **linear_page** and other for **multi_page.** Tar and gzip the parent directory using the command **tar -zcvf 18xxxxxxx.tar.gz** 18xxxxxxx to produce a single compressed file of your submission directory. Submit this tar gzipped file on Moodle.