

## Advanced Operating Systems - Homework 1

### **Motivation:**

You have learned a number of page replacement algorithms. Homework 1 asks you to implement some of them and propose your own idea.

### **Specification:**

1. Reference string: 1~600
2. Number of memory references: At least 60,000 times
3. Number of frames in the physical memory: 10, 20, 30, 40, 50, 60
4. Two test reference strings:
  - (1) Random: Arbitrarily pick one number for each reference.
  - (2) Locality: Simulate function calls. Each function call may refer a subset of  $1/12 \sim 1/6$  string (the length of string can be random).
5. Hardware-supported bits: Reference bit and modify bit
  - (1) Hardware will automatically set these two bits.
  - (2) There is a probability  $p$  that a page in the memory will be modified. The value of  $p$  will be 0.05, 0.1, 0.15, 0.2, and 0.25.

### **Requirements:**

1. You need to implement THREE algorithms for comparison:
  - (1) FIFO algorithm
  - (2) Optimal algorithm
  - (3) Enhanced second-chance algorithm
2. In addition, you should develop your own algorithm (not in the textbook). Your algorithm should at least win the FIFO one (in terms of page fault rate or cost). Of course, the more algorithms you win the better. There are two types of costs: 1) the number of interrupts and 2) the number of frames written to the disk. For 1), you need to clarify what situation will cause an interrupt in your simulations. For 2), when the modify bit of a frame is not set but it is replaced by the algorithm, the cost is zero since you need not write it to the disk.
3. For each algorithm and reference string, your report should include the following two figures:
  - (1) The relationship between page faults and the number of frames.
  - (2) The relationship between the costs (i.e., interrupt and writing) and  $p$ .In addition, your report should give some discussions about the behaviors of these algorithms.

4. You need to demonstrate your program to TA and submit your report in class.

**Due Day:**

2015/10/29

**Grading Policy:**

Programming 65%

Report 35%