

CS 532 HW 3
DUE Tuesday Feb 4 before 6pm

Part I – Programming Exercise: The Memory API

Note: In this exercise you are asked to write code that is NOT CORRECT. Please indicate this in one or more comments right in the code, to avoid any confusion in future.

A. Write a C program called MyBad.c that contains a memory leak and use the "free" utility to check the impact on your system. (hint: `man free`)

Specifically

1. write a program containing a for loop that allocates a buffer, forks a child process, then enters an inner for loop that takes some time
2. Start your code then use the free utility to note the changes in free space

SUBMIT your code MyBad.c

B. Copy MyBad.c to a new file called LessBad.c that adds one or more calls to `free()` to correct the memory leak.

SUBMIT your code LessBad.c

C. Write a C program called YourBad.c that contains a *buffer overflow*.

Specifically

1. Write a program with a command line argument that is a string of length 10
2. In your code copy the argument into a buffer of size 8
3. What is the result?

SUBMIT your code YourBad.c

Part 2 – Simulator Exercises: Paging

Exercises 1-4 in the textbook on pages 14-15 at the end of chapter 18 (Paging: Introduction).

Part 3 - Short Questions

A. In the *base and bounds* method for mapping virtual addresses to physical addresses, one register holds the base address that is added to each virtual address. Why do we need the bounds (also called "limit") register?

B. Why isn't *base and bounds* sufficient? (What problem(s) doesn't it solve?)

C. What is the difference between external and internal fragmentation?

D. Does the TLB improve performance because of temporal locality, spatial locality, or both?