# The Heap
# Activity Instructions

(format borrowed from TIDES and FIC workshops:)

---

Now we're thinking about the heap, and how dynamic memory allocation interacts with the rest of the system. Let's assume we are running a C-program that explicitly manages memory (could it do something else?) using a free-list allocator. The system is also demand-paged.

1. Write a 7-10 line program that includes the following: global data, dynamically allocated data, local variables, and a system call. We're good programmers here, so please also deallocate your data!

2. Where are the variables allocated? Do any of the variables point to storage in the heap? Which ones?

3. Now trace the steps of the main thread as it executes the program---including any steps into the OS or user-level libraries. At each step, consider:
    a. Is there a page fault? What happens?
    b. Is the thread in user mode or kernel mode?
    c. If there is a heap, what is the state of the heap? What is free? Where is data allocated?
    d. Is there a chance that the process will block or otherwise change state when executing that statement?