91.308 Operating Systems, November 4, 2014
Project #5

- This assignment is due on **Due Tuesday, November 25**.

1. **All** of your submissions must include a minimum of **four** separate files:
    - **File 1:** A **write-up** that first specifies what you think your **degree of success** with a project is (**from 0% to 100%**), followed by a brief discussion of your approach to the project along with a **detailed description** of any problems that you were **not** able to resolve for this project. Your write-up **must contain a summary** of your results. The **write-up** for this project is the **most important part of the project**, and must include the **summary results** discussed in class and detailed below. **Failure to specifically provide this information will result in a 0 grade** on your assignment. If you do **not disclose** problems in your write-up and problems are detected when your program is tested, you will receive a grade of 0.
    - **File(s) 2(a, b, c, …)**: Your **complete source code**, in one or more **.c** and/or **.h** files
    - **File 3**: A **make file** to build your assignment. This file must be named **Makefile**.
    - **Files 4(1-6):** Files that include your **resulting output** runs from each algorithm and memory size used in your project. This is a simple text file that shows your output, but make sure that you annotate it so that it is self descriptive and that all detailed output is well identified.

2. The files described above should be the only files placed in one of your subdirectories, and this subdirectory should be the target of your submit command (see the on-line file **Assignment_Submit_Details.pdf** for specific directions).

3. The program you will write will read a file which contains memory allocation requests and memory free operations. The allocation and free operations will be made against an initially empty fixed sized free memory pool. The program must support three allocation policies:

    - First fit linked list
    - Best fit linked list
    - Buddy system power of 2 block allocation (using a minimum allocation of 32 bytes)

- Your program must:

1. Accept three command line arguments, such that argv[1] is the **policy** to use, argv[2] is the **total memory free pool size** and argv[3] is the **name of the file** containing the allocation and free requests. The memory size of the total free pool for the main test cases will be **1 MByte,** and **512KB**. You may use other sizes in any additional experiments you attempt, but make sure you **describe** your configuration and results in your write-up.

2. Read each line of the input file in argv[3] and carry out one of the following actions:

    - Make a memory allocation if enough memory is available to satisfy an allocation request.

    - Return memory to a linked list and carry out any coalescing (buddying up) required.

    - Refuse the allocation only if there is not enough memory to satisfy it in any available free partition. Refused requests are remembered, so that you know to take no action when you see the corresponding free operation in the input stream. (For each alloc there is a free.)

3. Generate **one line of output for each line of input**, providing information about which request this is, how the request was handled, what the total amount of free space after the request is, and what the size of the largest free partition after the request is. Sample input and output files are shown below.

4. So you should **submit 6 files**, each containing your output for the **1MB and 512KB** cases for all three algorithms, and a final **summary in your write-up** that provides the results of each test in a format similar to:
    - First Fit 1MB: Total Allocations 490 of 500
    - First Fit 512KB: Total Allocations 374 of 500
    - Best Fit 1MB: Total Allocations 492 of 500

    Etc…..

```
INPUT FORMAT:
          SERIAL-NUM      REQUEST         SIZE       SERIAL-MATCH
              1            alloc          20000
              2            alloc         100000
              3            alloc           5050
              4            free                              2
              5            alloc          70500
              6            free                              1
              7            alloc         400000


OUTPUT FORMAT:
    MANAGEMENT POLICY = First Fit      POOL SIZE = 512 KB

SERIAL-NUM    REQUEST    SIZE   ALLOC-ADDR   TOTAL-FREE   LARGEST-PART
      1        alloc     20000           0     504288        504288
      2        alloc    100000       20000     404288        404288
      3        alloc      5050      120000     399238        399238
      4        free     100000                 499238        399238
      5        alloc     70500       20000     428738        399238
      6        free      20000           0     448738        399238
      7        alloc    400000          -1     448738        399238
```

- The data file you must use for this project is available in
  ~bill/cs308/proj5_data

  or at the website

  www.cs.uml.edu/bill/cs308/proj5_data.