EECS 111 - System Software

Spring 2010

Assignment 5

Posted: Tuesday, May 18, 2010

Due: Tuesday, May 25, 2010, 12pm (noon)

A. Discussion: Main Memory Management

The goal of these exercises is to review and clarify the understanding of essential aspects covered in recent lectures.

- Memory management: Exercises 8.1, 8.3, 8.4, and 8.5.
- Paging: Exercises 8.17 and 8.18

These topics are planned to be discussed in this week's discussion session.

B. Project: Memory Allocation Strategies

The goal of this project assignment is to examine various memory allocation strategies by implementing the first-fit, best-fit, and worst-fit algorithms.

Step 1: Setup

- Start from the provided template file **`memalloc_template.c`**. Name your source file **`memalloc.c`**.
- Read the template code carefully first, then compile and run it to understand the given setup. When compiling, you can choose one of the three algorithms by defining the corresponding preprocessor definition. For example, to select the first-fit algorithm, use the option **`–DFIRSTFIT`** as compiler option for **`gcc`**.
- For this assignment, we will only modify the implementation of the three functions **`firstFit()`**, **`bestFit()`**, and **`worstFit()`**. All other parts in the program template should remain unmodified.

Step 2: Implement the memory allocation algorithms

Implement the three basic memory allocation algorithms described in Chapter 8 of the text book in the bodies of the three functions **`firstFit()`**, **`bestFit()`**, and **`worstFit()`**. Each function should allocate the requested **`size`** of memory in the global **holes** array and return the index of the hole selected. In the case the requested memory cannot be allocated, the function should return **-1**.

Step 3: Compile and test your implementation

When running your program, your execution log should show the list of available memory holes at every step, as well as the index of the hole that actually got selected (if the allocation was successful).

In a text file **memalloc.txt**, list the number of satisfied requests for each of the three algorithms. Compare the results and briefly explain why or why not the observed results match your expectation.

Finally, consider the "best" algorithm. In particular, is it possible that the first-fit algorithm beats the best-fit algorithm in certain situations? If you think yes, provide a specific example. If you think no, explain why not.

Deliverables:

1. Statement: "I have read the Section on Academic Honesty in the UCI Catalogue of Classes (available online at http://www.editor.uci.edu/catalogue/appx/appx.2.htm#gen0) and submit this work accordingly."
2. Source file **memalloc.c**, execution log **memalloc.log**, and a brief text file **memalloc.txt** explaining the results and answering the above question (2 paragraphs) [100 points].

Submission instructions:

To submit your homework, send the deliverables in an email with subject "EECS111 HW5" to the course instructor at doemer@uci.edu.

To ensure proper credit, be sure to send your email before the deadline:
Tuesday, May 25, 2010, at 12:00pm (noon).

--

Rainer Doemer (EH 3217, x4-9007, doemer@uci.edu)