CS33 Homework Assignment 8

Due: 11:59pm, November 11, 2022

1. In lecture 24 we discussed how linking takes place and, in particular, how it works using ELF. Slide XXIV-37 gives the relocation section and the object code for *subr*. Show the modified object code after it has been relocated using the information given in slide XXIV-41. (Keep in mind that the architecture is little-endian; be sure to read the notes that go along with Slide XXIV-41.)

2. *Malloc* and *free* were designed to work well in situations in which not much, if anything, is known about the pattern of allocation and free requests that a program might make. Because the sizes of requests are not known in advance, there can be significant fragmentation issues as well as time wasted due to searching for sufficiently large free blocks. But suppose a program dynamically allocates memory to accommodate variables of just, say, six datatypes — u_t, v_t, w_t, x_t, y_t, and z_t, each of a different size (assume all sizes are relatively small and relatively prime). Though it's still the case that the order of allocation and free requests is essentially random, explain how one might be able to exploit the fact that all requests are from the six sizes so that allocation and freeing is much more efficient than it would be if they were done strictly with malloc and free (in particular, there are no external fragmentation issues). Your approach should be built on top of *malloc* and *free*, i.e., it calls them as subroutines. You don't need to provide code, but simply explain the approach.