University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS 61C, Fall 2003

Lab 3: Malloc

Goals

This lab will give you practice working with the malloc function.

Initial preparation

Read sections 6.1–6.5 in K&R.

Copy the directory ~cs61c/labs/lab03 to your home directory.

Working with partners

If you work with a partner on these exercises, make sure that you both understand all aspects of your solutions.

Background

The code in lists.c supplies a partial implementation of a data type representing a list of names; its corresponding header file is lists.h. (Note that the #include statements surround the file name with double quotes rather than brackets.) A program that tests the list functions is provided in listtest.c; compile the pair of .c files into a program with the command

```
gcc lists.c listtest.c
```

The listtest.c program reads successive lines from the input without prompting, replacing the terminating newline with a '\0'. Each input string is added to the front of the list being accumulated. The program doesn't prompt you individually for the names to add to the list; merely type ^D (control-D, which means end-of-input) at the start of a line to indicate that no more strings are to be added to the list, and the names will be printed one per line in the order that they appear in the list.

Exercise 1

Define struct `ListNode`. Your solution should not limit the length of a name, even though the listtest.c program does.

Exercise 2

Complete the `cons` function and test it with the listtest.c code. Don't change anything else in any of the files.

Exercise 3

Invent an exam question based on this lab assignment that would test a student's understanding of the malloc function and when it is to be used.

Contents of ~cs61c/labs/lab03

lists.h

```
struct ListNode;

struct ListNode * newList ( );
struct ListNode * cons (char *, struct ListNode *);
void print (struct ListNode *);
```

lists.c (to which you add code)

```
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include "lists.h"

struct ListNode ...           /* You supply this definition. */

/* Return an empty linked list. */
struct ListNode * newList ( ) {
   return 0;
}

/* Return the result of adding the given string to the front of the
given list. */
struct ListNode * cons (char * s, struct ListNode * list) {
   /* You supply the body of this function. */
}

/* Print the names in the given linked list, one per line. */
void print (struct ListNode * list) {
   if (list != 0) {
      printf ("%s\n", list->name);
      print (list->next);
   }
}
```

listtest.c

```
#include <stdio.h>
#include "lists.h"

int main ( ) {
   char word[100];
   printf ("Collecting names into a list.\n");
   struct ListNode * list = newList ( );
   while (gets(word)) {
      list = cons (word, list);
   }
   printf ("Printing the names in the list.\n");
   print (list);
   return 0;
}
```