CSE 320 Spring 2013

Computer Project #10 -- SPARC Data Organization

Assignment Overview:

This assignment develops familiarity with processing data structures in assembly language. You will develop the SPARC assembly language functions to complete a program to manage statistics for a basketball team.

It is worth 40 points (4% of course grade), and must be completed no later than 11:59 PM on Thursday, April 18.

Assignment Specifications:

1) The program will use an ordered table to maintain the data set, where each player's jersey number will serve as a unique key to identify that player. The capacity of the ordered table will be determined when it is created.

2) The instructor-supplied driver module (function "main" and associated functions) will perform all input and output, and will manage the overall operation of the program.

3) You will supply the functions whose declarations are listed below:

```
int search( struct table*, unsigned long, struct player** );
int delete( struct table*, unsigned long );
int insert( struct table*, unsigned long, char*, int, int, int, int );
```

Those functions (as well as any other functions you choose to develop) will be implemented in assembly language.

Assignment Deliverables:

The deliverables for this assignment are:

```
proj10.makefile  -- the makefile which produces "proj10"
proj10.support.s -- the source code for your support module
```

Be sure to submit your files for grading via the "handin" program.

Assignment Notes:

1) The file "project10.interface.h" (appended below) includes all relevant declarations, along with descriptive comments.

2) The file "project10.driver.o" contains the instructor-supplied driver module.

3) The file "project10.data" contains a sample data set (the statistics for the MSU Women's Basketball team during the 2012-2013 season). Your program must function correctly for that sample data set, as well as any other properly formatted data set.

4) You may wish to create stubs for the required functions, then translate, link and execute the program to explore the behavior of the driver module.

```
/*********************************************************************/
/*  Declarations for Project #10                                     */
/*********************************************************************/


struct player
{
  unsigned short number;    /* player's jersey number (key)   */
  char name[25];            /* player's name                  */
  unsigned short games;     /* number of games played         */
  unsigned short bask2;     /* number of 2-point baskets made */
  unsigned short bask3;     /* number of 3-point baskets made */
  unsigned short free;      /* number of free throws made     */
  unsigned short points;    /* total points scored            */
  float points_per_game;    /* points per game played         */
};


struct table
{
  unsigned short capacity;  /* number of elements in table    */
  unsigned short count;     /* number of players in table     */
  struct player* memory;    /* pointer to array of players    */
};


/*********************************************************************/
/*  Function:  search                                                */
/*                                                                   */
/*  Purpose:  locate and return a pointer to a player, if the        */
/*  player is present in the table.                                  */
/*                                                                   */
/*  Arguments:                                                       */
/*    pointer to table of players                                    */
/*    jersey number of player to be located                          */
/*    pointer to pointer to player                                   */
/*                                                                   */
/*  Return value:                                                    */
/*    1 (true) if player located, 0 (false) otherwise                */
/*********************************************************************/

int search( struct table*, unsigned long, struct player** );
```

```
/*********************************************************************/
/*  Function:  delete                                                */
/*                                                                   */
/*  Purpose:  delete a player from the table, if the                 */
/*  player is present in the table.                                  */
/*                                                                   */
/*  Arguments:                                                       */
/*    pointer to table of players                                    */
/*    jersey number of player to be deleted                          */
/*                                                                   */
/*  Return value:                                                    */
/*    1 (true) if player deleted, 0 (false) otherwise                */
/*********************************************************************/

int delete( struct table*, unsigned long );


/*********************************************************************/
/*  Function:  insert                                                */
/*                                                                   */
/*  Purpose:  insert a player into the table, as long as there is    */
/*  room in the table and the player is not already present.         */
/*                                                                   */
/*  Arguments:                                                       */
/*    pointer to table of players                                    */
/*    jersey number of player to be inserted                         */
/*    pointer to name of player                                      */
/*    number of games played                                         */
/*    number of 2-point baskets made                                 */
/*    number of 3-point baskets made                                 */
/*    number of free throws made                                     */
/*                                                                   */
/*  Return value:                                                    */
/*    1 (true) if player inserted, 0 (false) otherwise               */
/*********************************************************************/

int insert( struct table*, unsigned long, char*, int, int, int, int );
```