C library function - mktime()

Description

The C library function **time_t mktime(struct tm *timeptr)** converts the structure pointed to by **timeptr** into a time_t value according to the local time zone.

Declaration

Following is the declaration for mktime() function.

```
time_t mktime(struct tm *timeptr)
```

Parameters

- **timeptr** − This is the pointer to a time_t value representing a calendar time, broken down into its components. Below is the detail of timeptr structure

```
struct tm {
   int tm_sec;         /* seconds,  range 0 to 59          */
   int tm_min;         /* minutes, range 0 to 59           */
   int tm_hour;        /* hours, range 0 to 23             */
   int tm_mday;        /* day of the month, range 1 to 31  */
   int tm_mon;         /* month, range 0 to 11             */
   int tm_year;        /* The number of years since 1900   */
   int tm_wday;        /* day of the week, range 0 to 6    */
   int tm_yday;        /* day in the year, range 0 to 365  */
   int tm_isdst;       /* daylight saving time             */
};
```

Return Value

This function returns a time_t value corresponding to the calendar time passed as argument. On error, a -1 value is returned.

Example

The following example shows the usage of mktime() function.

```
#include
#include

int main () {
   int ret;
   struct tm info;
   char buffer[80];

   info.tm_year = 2001 - 1900;
   info.tm_mon = 7 - 1;
   info.tm_mday = 4;
   info.tm_hour = 0;
   info.tm_min = 0;
   info.tm_sec = 1;
   info.tm_isdst = -1;

   ret = mktime(&info);
   if( ret == -1 ) {
      printf("Error: unable to make time using mktime\n");
   } else {
      strftime(buffer, sizeof(buffer), "%c", &info );
      printf(buffer);
   }

   return(0);
}
```

Let us compile and run the above program that will produce the following result −

```
Wed Jul 4 00:00:01 2001
```