Operating Systems lab 3

Robin Sommer, s2997592
Suzanne Maquelin, s26963200

March 2018

To implement our system call UTCtime to make up for leap seconds that are caused by a change in the orbit of the earth, we have taken the following steps:

1. Adding an entry to the PM server system call table.
2. Adding a call number definition for the table entry.
3. Define the system call's function prototype.
4. Create the function for the prototype.
5. Adding a wrapper for the new system call.

Adding an entry to the system call table

The first step was to create a new entry to the system call table. We have done this by taking an existing entry that had no usage yet. We have picked number 35. The changed line: *do_utctime, /* 35 = utctime */*

Adding a call number definition for the table entry

The second step was to add our call number to the callnr.h, so our entry would be recognized by the system. The changed line: *#define UTCTIME 35;*

The system call's prototype

The third step included adding the prototype of our function do_utctime() in the proto.h. The changed line: *int do_utctime(void);*

The function

In the fourth step we implemented the function do_utctime() in the existing time.c. Thus we did not make changes to the Makefile. For this function we used computed the time in the same way as the regular time function. Then we added the appropriate amount of leap seconds to it. This is hardcoded, since the leapseconds are unpredictable and cannot be calculated.

The wrapper

For the wrapper the idea is to make an easy-to-use function that returns an integer. The system call itself uses messages and it is not trivial to retrieve the result from it. We have added the function utctime() to the standard library. For this we have add a function protype to unistd.h (*int utctime();*) and created the c file utctime.c, which contains the implementation. We then added utctime.c to the Makefile associated with the standard library.