Here's an example of filling up the time sheet. It records the time that I spent in implementing the example program to be used in the lectures on 12 and 13 September, 2001 (Lex & Yacc and Systematic Software Development). The task at hand was very simple. Design and implementation of a simple calculator (with variables and assignments to variables). Did not require any overall design since this example is discussed in many books including *The Unix Programming Environment* (Kernighan and Pike) and *Compilers: Principles, Techniques, and Tools* (Aho, Sethi, and Ullman) as also the original yacc manual by Johnson. However, this code is completely mine, I did not copy it from anywhere :-)

There are basically three modules : scanner, parser, and symtab handler. I used a simple linked list for storing names and their values. The code was so simple that no paper design was required either.

Use of invariants and use of conditional compilation for simultaneous testing (without changing the code) was a great help.

---

**Time Recording Log, CS699 (Software Laboratory), Dept. of CS&E, IIT Bombay**

Name : Uday Khedker    Roll No: ____________

Login name (for email) : uday

Following are some examples of activities for which the time should be recorded. Feel free to include any other activity which requires time to be spent toward this course. Specify the program unit wherever applicable (in terms of module, program file, data structure/class, function etc.)

| | |
|---|---|
| Reading/understanding the assignment statement | Reading documents/books/manuals |
| Discussing with others | Designing overall solution (on paper) |
| Designing the program on paper | Actual Coding Time |
| Testing/Debugging | Modifying/Cleaning up the program |

C? Stands for whether the work was completed or not.

| Date | Start Time | Stop Time | Interruption | Time Spent | Activity | Comments | C? |
|---|---|---|---|---|---|---|---|
| 11/10 | 11:55 | 13:20 | 5,5,10 | 1:05 | Symtab | Used simple linked list. | √ |
| | | | | | | Use of invariant save a lot of time. | |
| | | | | | | Bottom up development and simultaneous | |
| | | | | | | testing using conditional compilation | |
| | | | | | | was a great help. | |
| 11/10 | 16:00 | 17:00 | 10,20 | 0:30 | Symtab | Made testing more robust. Had to refer | √ |
| | | | | | | to K&R for scanf in the case of invalid | |
| | | | | | | input. | |
| 11/10 | 17:30 | 19:30 | 10,5,15 | 1:30 | Scanner | Very simple. Only three REs. | √ |
| | | | | | | Independent testing was a great help. | |
| | | | | | | Getting the macro 'token' to work | |
| | | | | | | correctly was very interesting and | |
| | | | | | | rewarding. | |
| 12/10 | 10:00 | 10:15 | | 0:15 | Makefile | | √ |
| 12/10 | 10:30 | 13:15 | 5,10,5,10 | 1:15 | Parser | Very simple, only three non-terminals | √ |
| | | | | | | and a few tokens. | |
| | | | | | | Had to struggle with \n in the scanner | |
| | | | | | | and generation of test version. | |
| 12/10 | 13:15 | 13:30 | | 0:15 | Makefile | Included test version generation of | √ |
| | | | | | | symtab and scanner in the makefile. | |