/\*

\* timer.c : contains two timing functions to measure process time

\*/

#include <sys/times.h>

#include <unistd.h>

#include <stdio.h>

#include "shell.h"

#include <bits/time.h>

/\* Storage for baseline times. \*/

static clock\_t start\_time;

static struct tms begin\_tms, end\_tms;

/\* Save a baseline of user and system CPU times, plus wallclock time. \*/

void set\_timer(void) {

//struct tms tmbuf;

/\* Fill in code. \*/

start\_time = times(&begin\_tms);

}

/\* Get second set of times, and take delta to get wallclock time. Display. \*/

void stop\_timer(void) {

//struct tms tmbuf;

clock\_t end\_time;

double ticks;

ticks = sysconf(\_SC\_CLK\_TCK);

/\* Get delta times and print them out.

\*

\* Fill in code. \*/

end\_time = times(&begin\_tms);

printf("real %lf\t",(end\_time - start\_time)/ ticks);

printf("user %lf\t",(end\_tms.tms\_utime - begin\_tms.tms\_utime) / ticks);

printf("sys %lf\n",(end\_tms.tms\_stime - begin\_tms.tms\_stime) / ticks);

/\*

int sc\_clk\_tck;

sc\_clk\_tck = sysconf(\_SC\_CLK\_TCK);

struct tms begin\_tms, end\_tms;

clock\_t begin, end;

system("date");

begin = times(&begin\_tms);

sleep(2);

end = times(&end\_tms);

printf("real time: %lf\n", (end - begin) / (double)sc\_clk\_tck);

printf("user time: %lf\n",

(end\_tms.tms\_utime - begin\_tms.tms\_utime) / (double)sc\_clk\_tck);

printf("sys time: %lf\n",

(end\_tms.tms\_stime - begin\_tms.tms\_stime) / (double)sc\_clk\_tck);

printf("child user time: %lf\n",

(end\_tms.tms\_cutime - begin\_tms.tms\_cutime) / (double)sc\_clk\_tck);

printf("child sys time: %lf\n",

(end\_tms.tms\_cstime - begin\_tms.tms\_cstime) / (double)sc\_clk\_tck);

return;

\*/

}