LAB1, Measuring the CPU Speed (voluntary due to testing problems)

Goal

Write a program that displays the clock rate of an Intel x86 CPU

Given files

Assuming you copied the lab files according to the lab setup instruction you can find the files you need in `~/TDDI11/lab1`. Your modifications goes to `main.c`. You can copy `main.c` from the `hello_world` example to get a starting point.

Assignment

Use two functions from the libepc library:

```
DWORD32 Now_Plus(int)
QWORD64 CPU_Clock_Cycles(void)
```

`Now_Plus(0)` returns a 32-bit integer. `Now_Plus(int n)` returns the 32-bit integer that Now_Plus(0) would return if called n seconds from the moment when `Now_Plus(int n)` is called.

Thus, call `Now_Plus(N)` first and store the result in a variable time-out. The first time `Now_Plus(0)` returns a number greater than or equal to time-out you know that N seconds have elapsed since you called `Now_Plus(N)`. Hence, you have a way to measure time.

Every x86 CPU chip has a 64-bit counter that increments at the processor clock rate. `CPU_Clock_Cycles` returns the value of the counter.

Based on `Now_Plus` you can measure a time interval. With the help of `CPU_Clock_Cycles` you can count how many clock impulses you got in the measured time interval. Thus, you may easily obtain the clock frequency.

Deliverables

Show your code to the assistant and demonstrate your application.

Obtaining CPU Clock Cycles correctly

Unfortunately `CPU_Clock_Cycles` does not seem to work correctly unless QEMU runs on a real Intel CPU. Since solaris run on Sparc and our available Linux server runs on AMD you must find your own solution as to how to test your solution.

Reference

Look in `/home/TDDI11/sw/include` to find all header files that declare functions you can use.