Task Manager

Contents

- Task manager
- Experiments with frequent real workloads
  - Word (printing a document)
  - Windows XP (start activities)
  - Internet access
  - Virus scan
  - Music player
  - Acrobat (making pdf file from doc file)
- Experiments with the flip-flop program

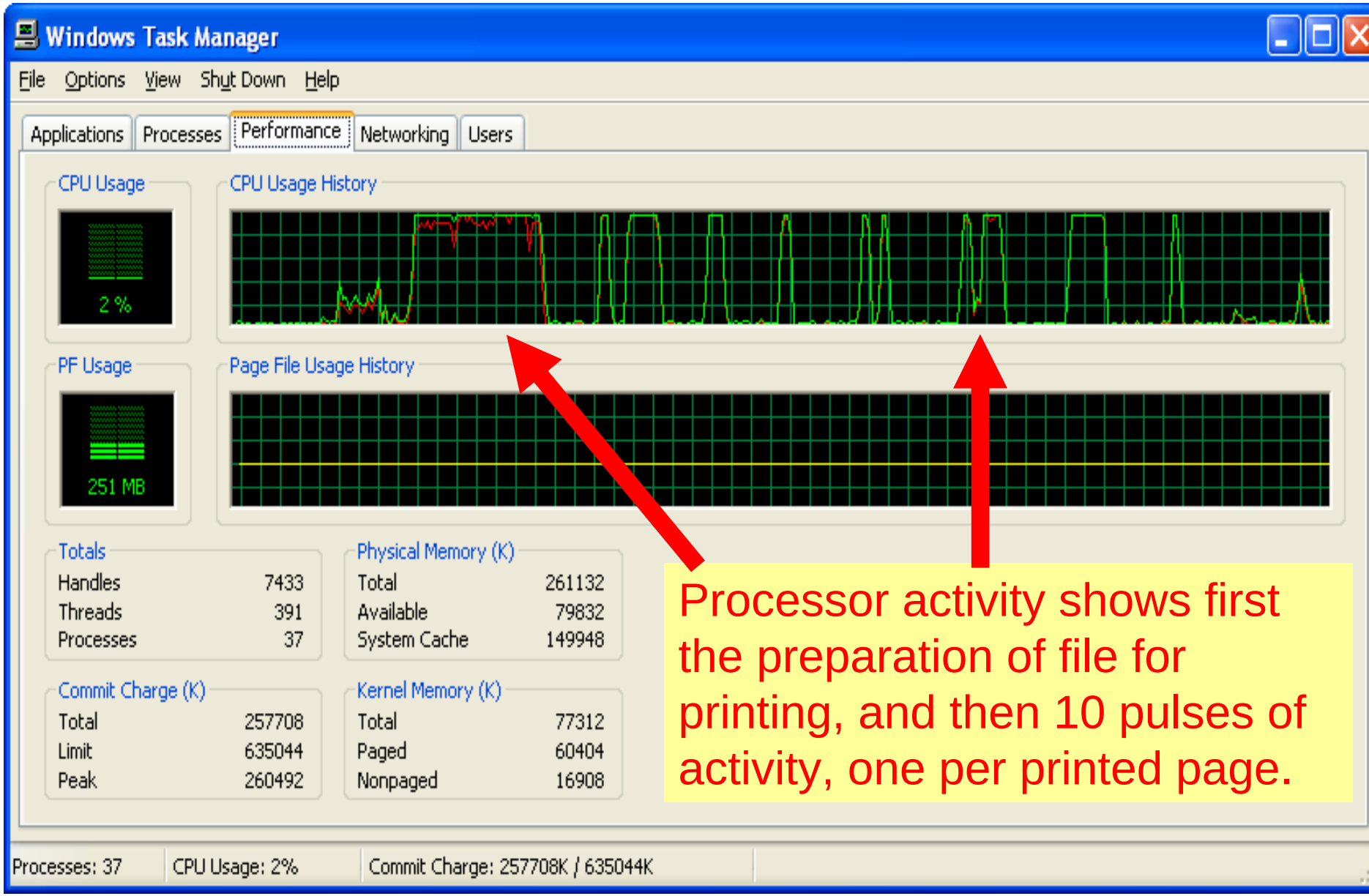
Task Manager

- Functions:
  - Tracking status of applications
  - Tracking all processes
  - Measuring usage of CPU and page file activity
  - Monitoring network activities
  - Monitoring users
- Activation of task manager:
  - Right click the taskbar
  - Start Task Manager

What Is the processor activity for printing a10 page document from WinWord?

- Printing is performed using a laser printer. We have no information to develop an analytic or simulation model for solving this problem.
- The only way to find the solution is to measure processor utilization using a suitable tool.
- Two measurement tools are available under Windows:
  - Task manager (simple)
  - Performance monitor (more complex)

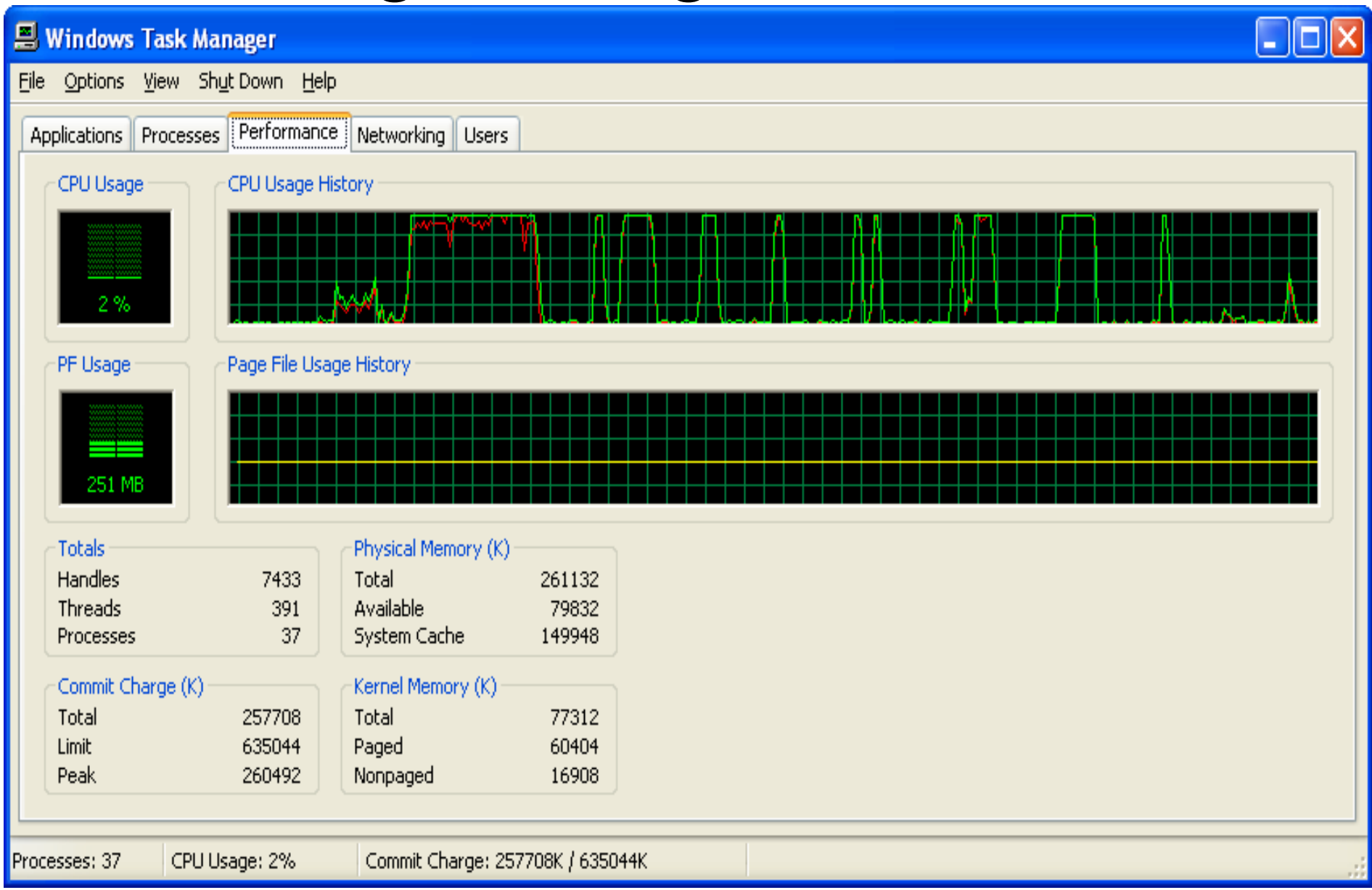
Printing 10 Pages from Word (Dell Dimension 8200)



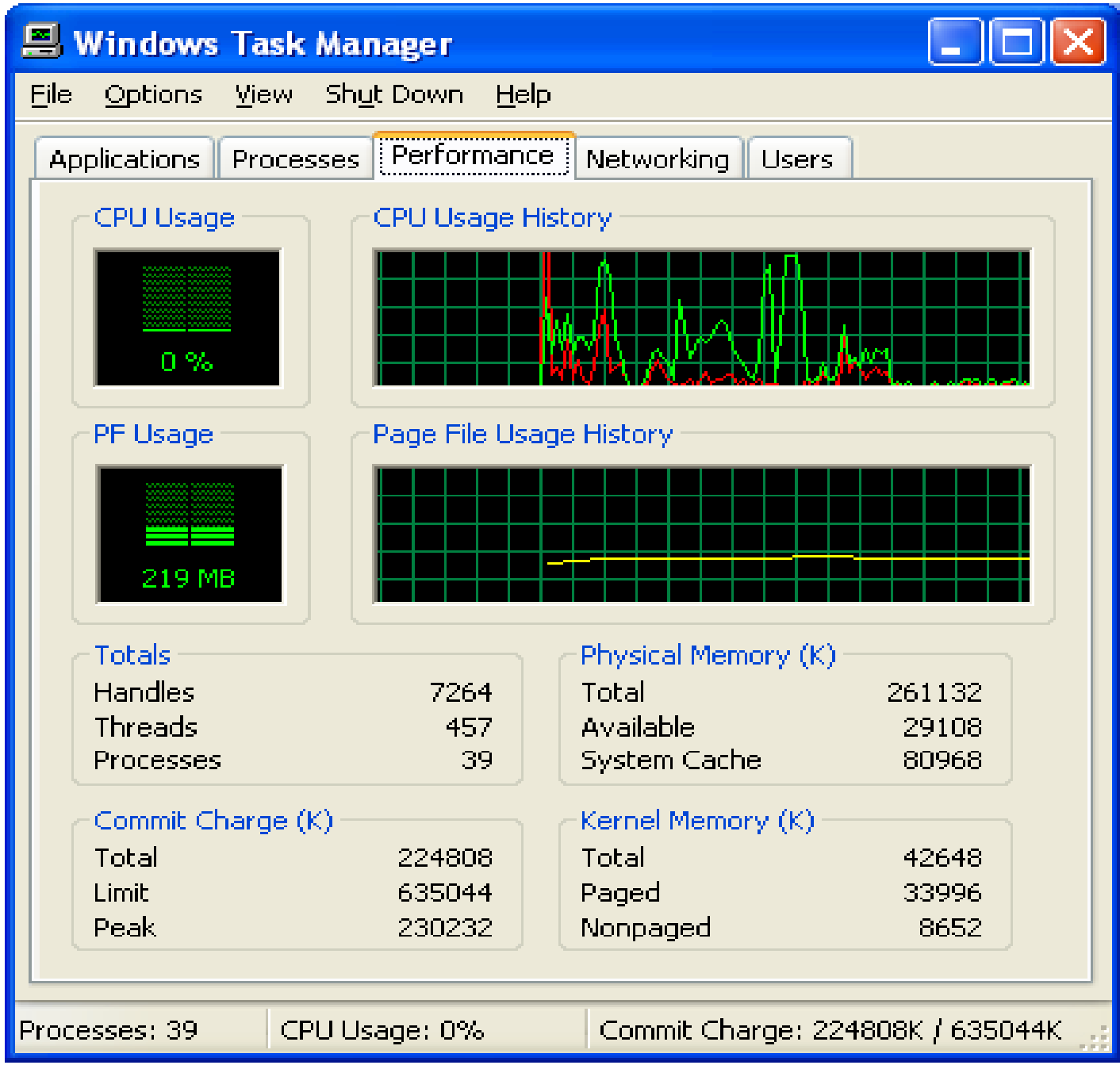
Observations

- This is an example of using CP to serve the I/O operations. This is typical for inexpensive machines where peripherals do not have advanced (and expensive) intelligent controllers.
- Most of the measured processor activity is in kernel mode (CP working for the OS).
- In advanced computer architectures I/O activities are performed with minimum use of CP.

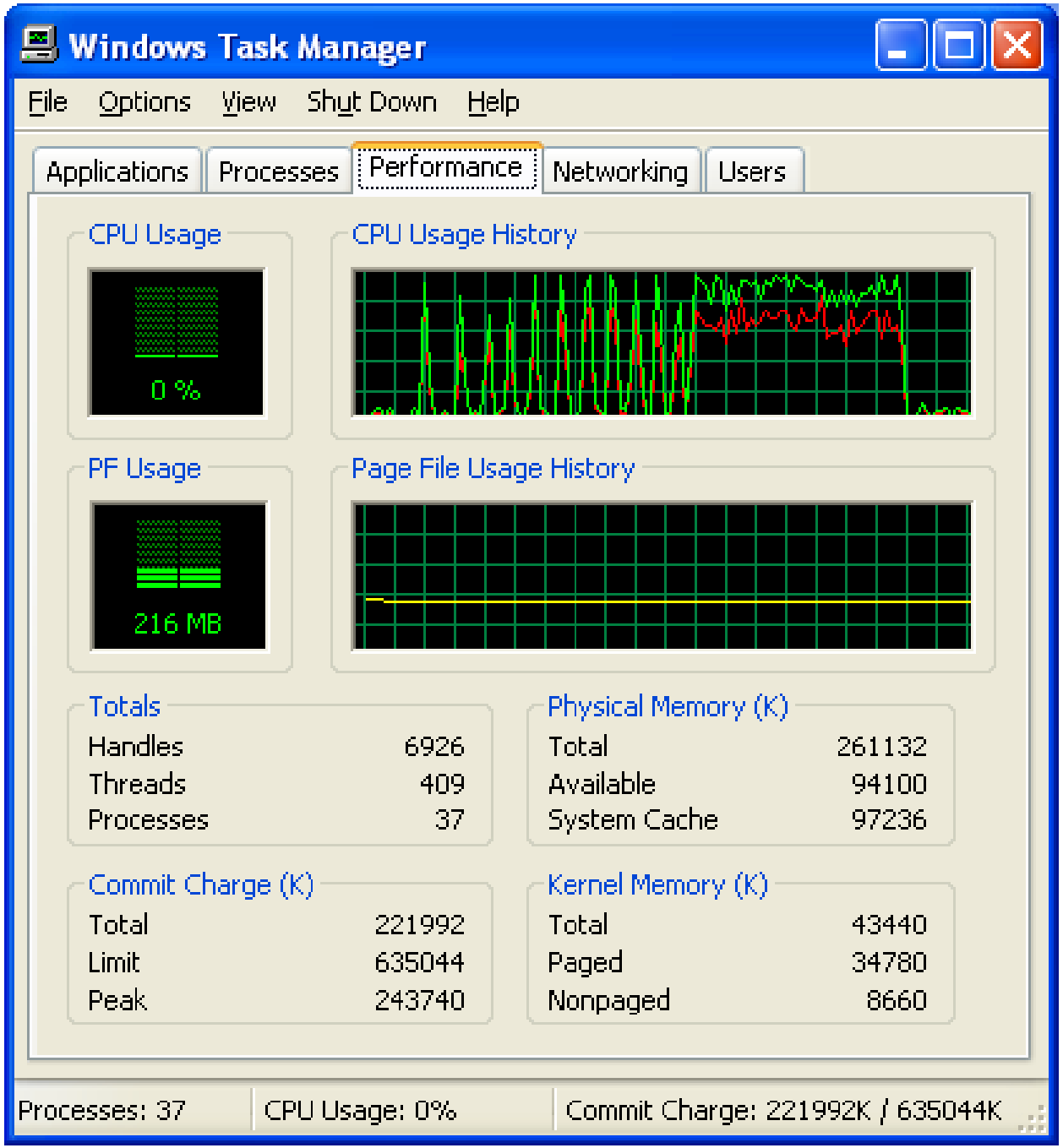
Printing 10 Pages from Word



Windows XP start activities (Dell Dimension 8200)



Moving task manager window from A to B followed by an interval of permanent move



Jozo Dujmović

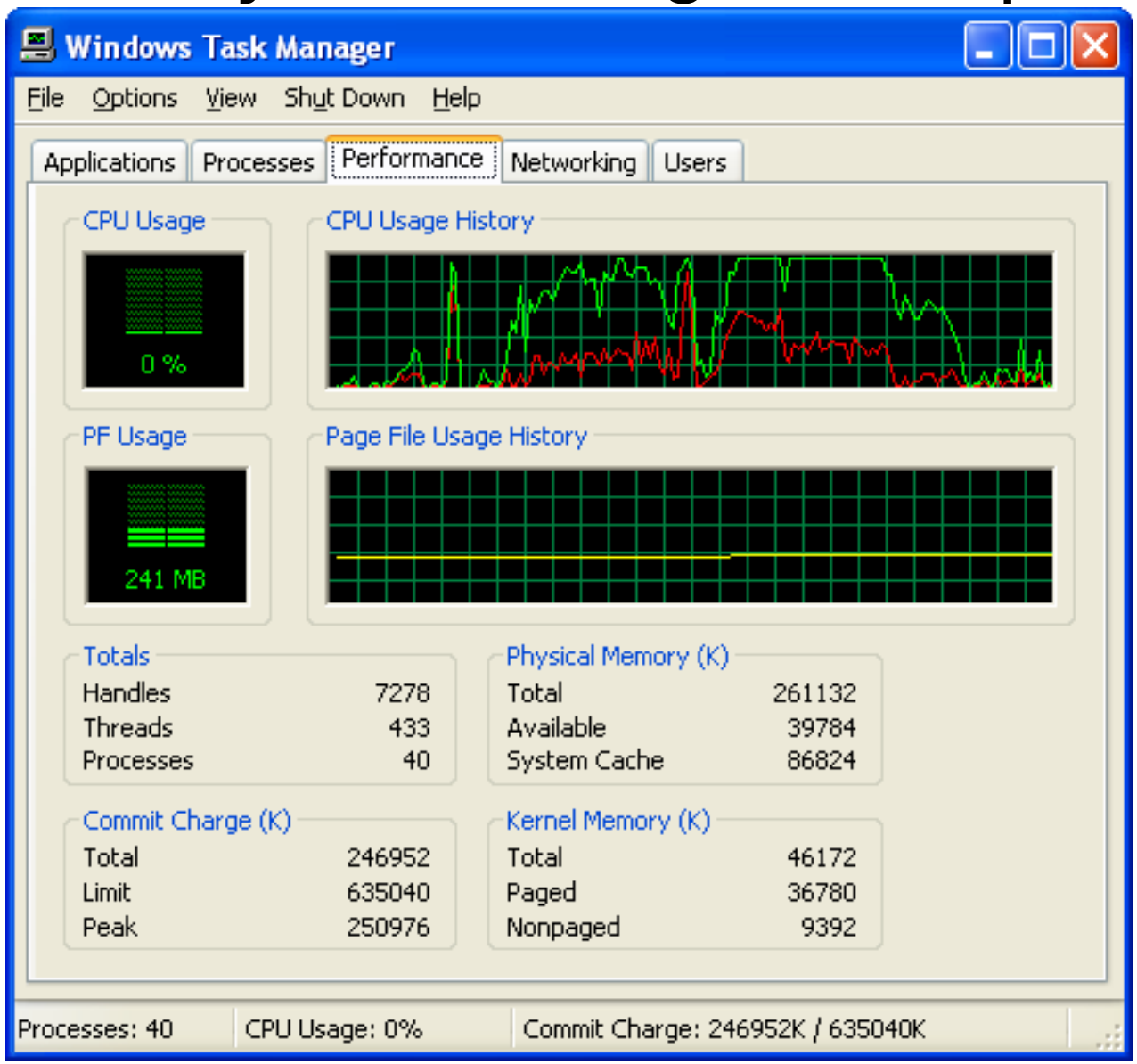
Processor Activity for starting Netscape

Computer = Dell Dimension 8200

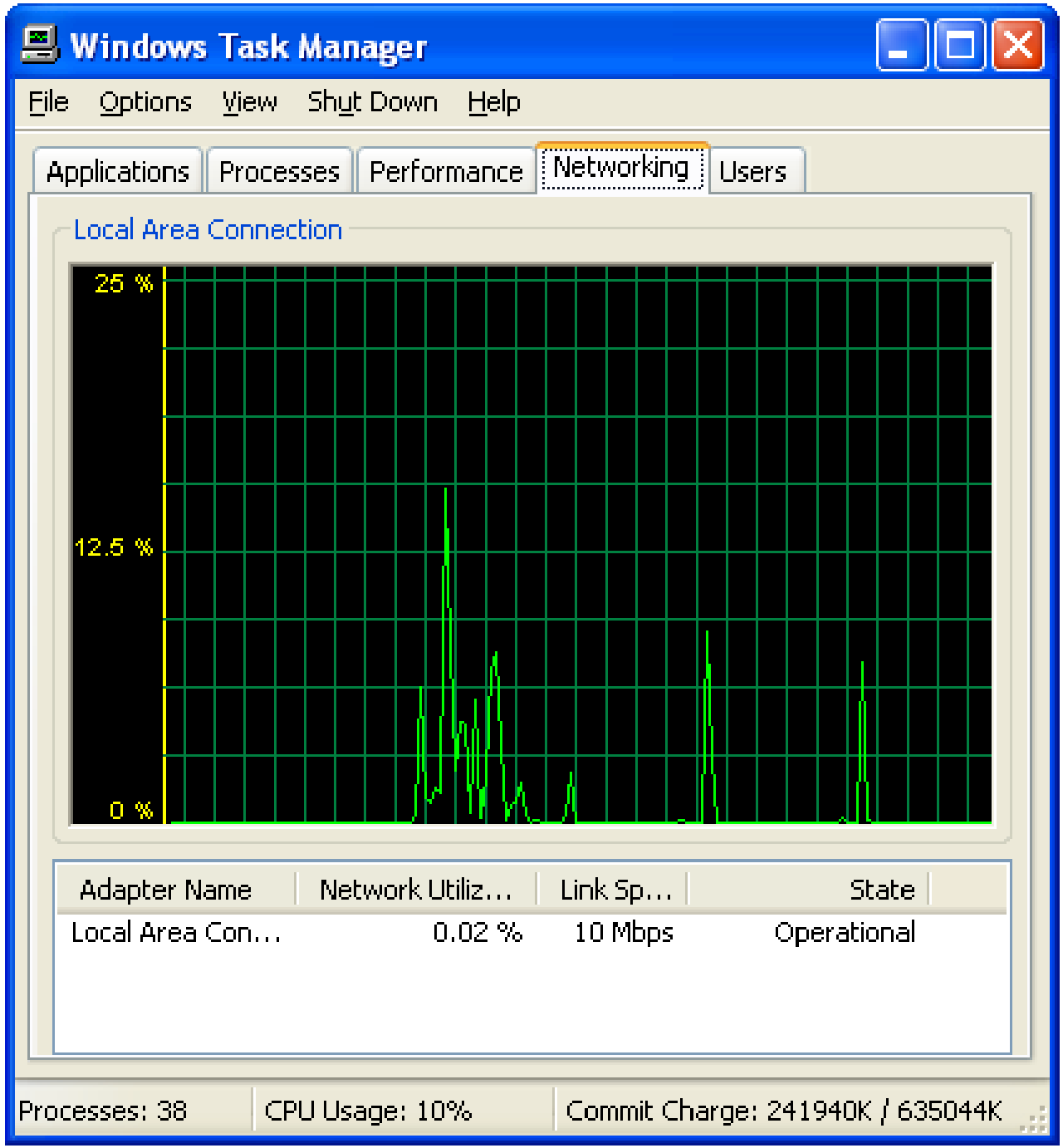
CPU = Pentium 4 @ 1.69 GHz

Memory = 256 MB of RAM

Workload includes a significant kernel activity.

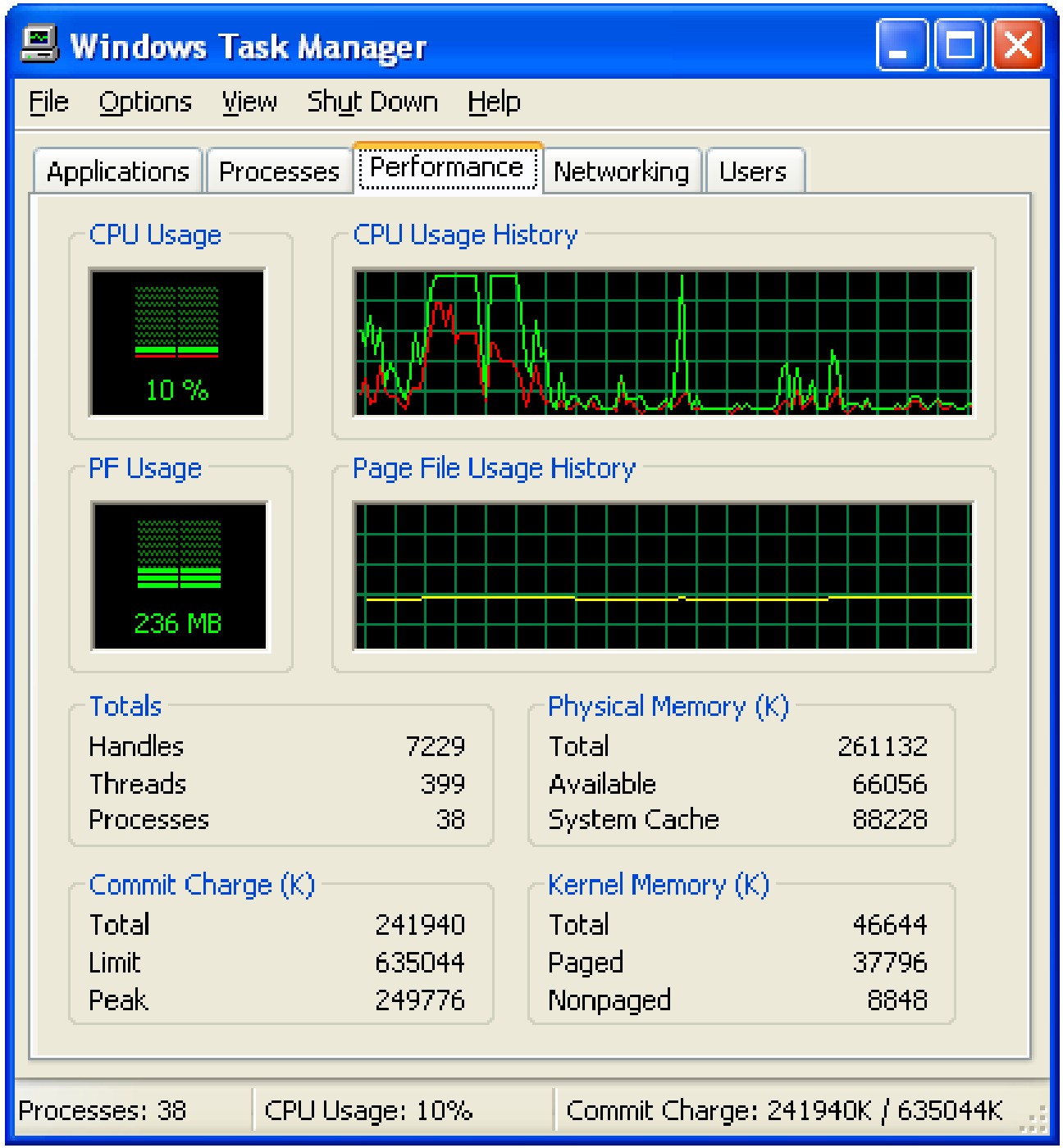


Starting Netscape followed by activating webmail and reading a message (networking activity)



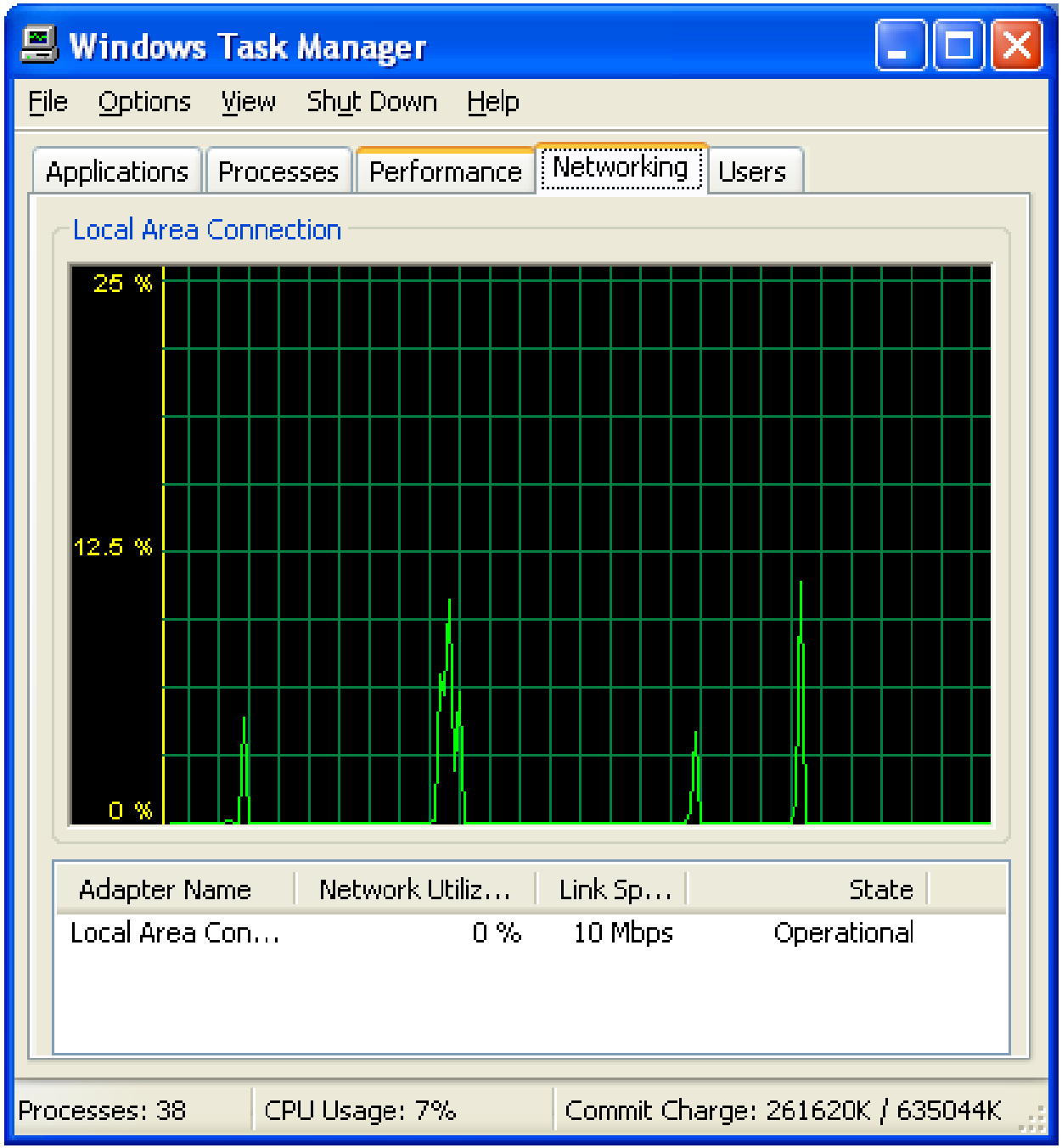
Jozo Dujmović

Starting Netscape followed by activating webmail and reading a message: CP activity, kernel+user

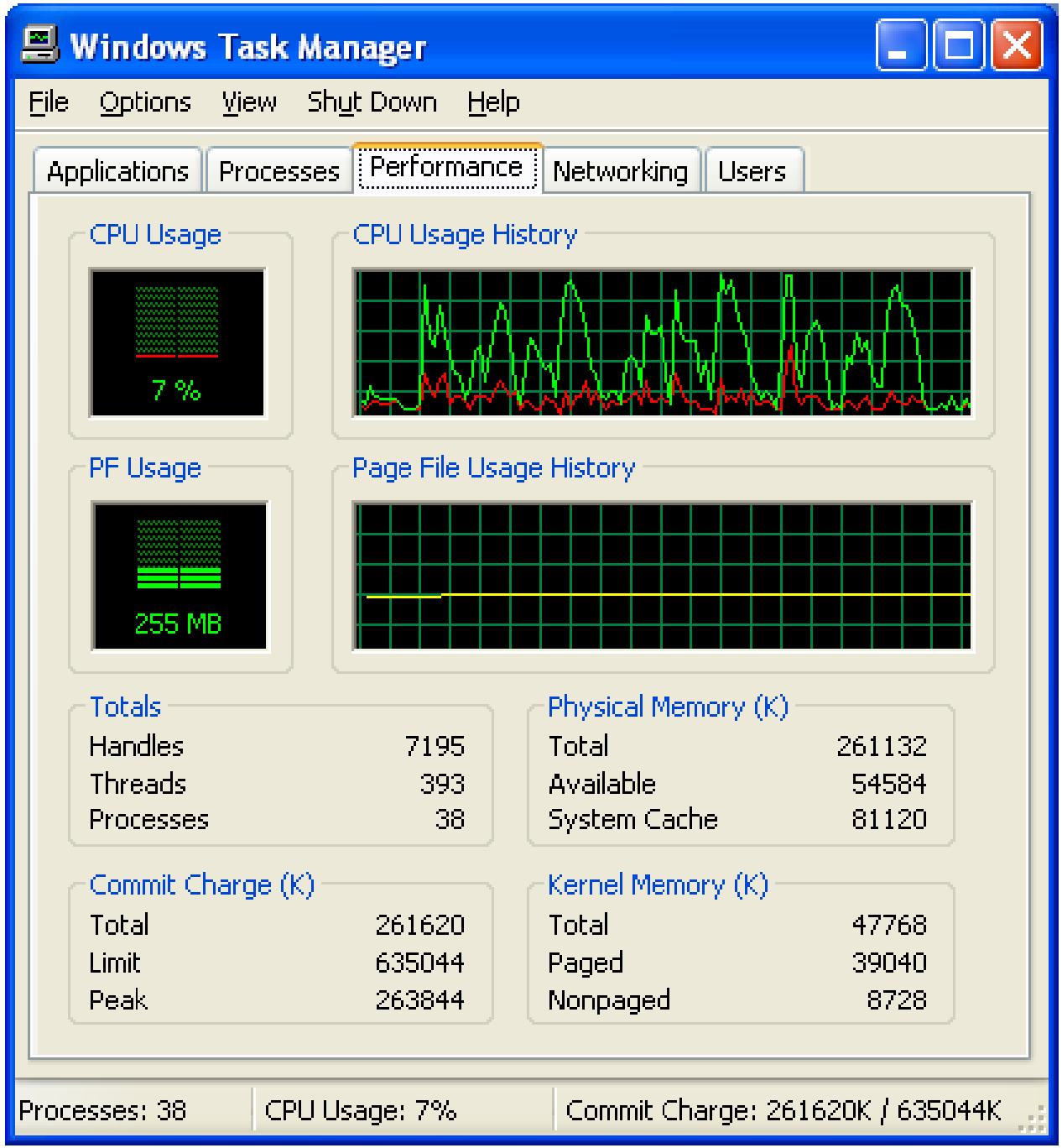


Jozo Dujmović

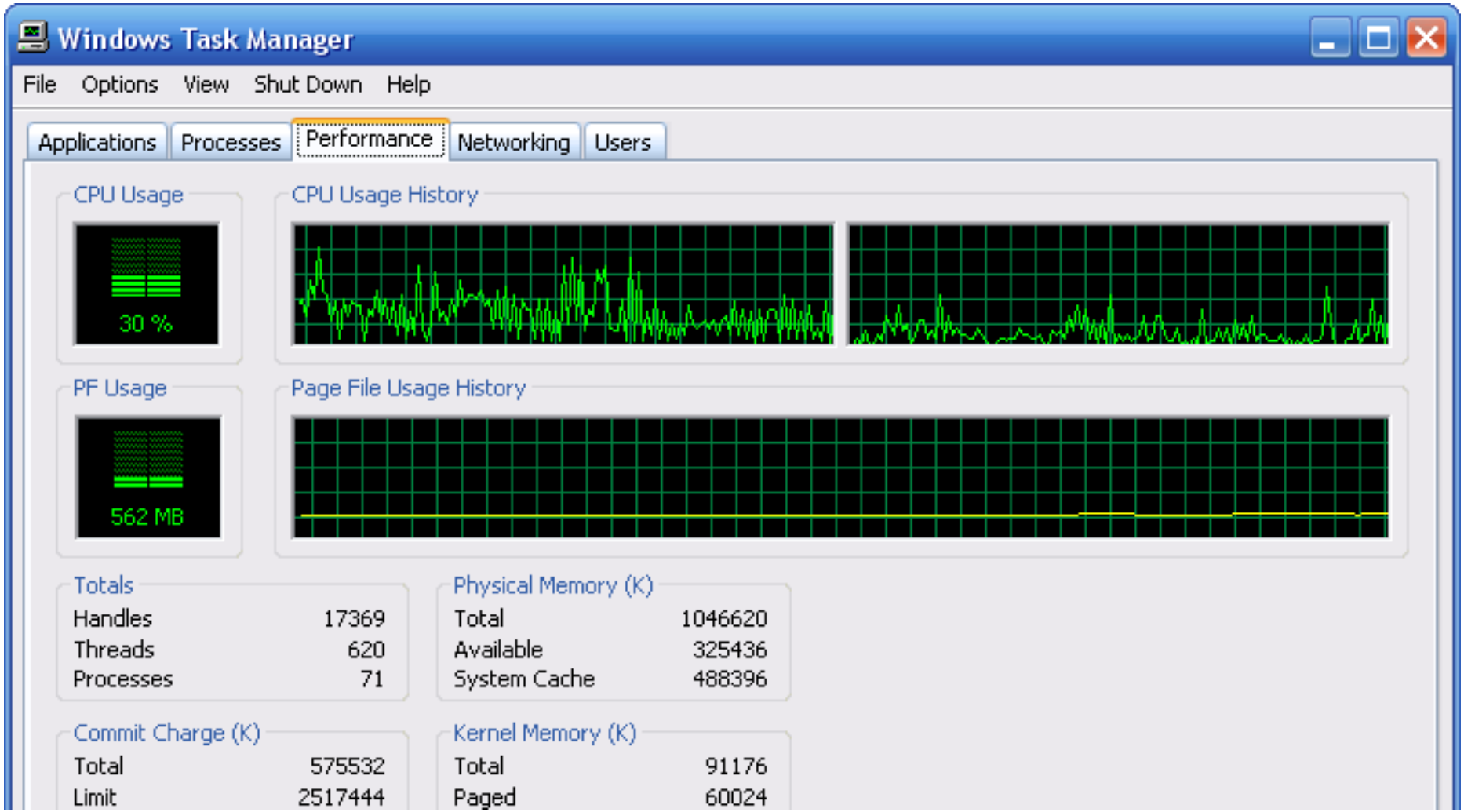
Internet access followed by viewing the retrieved contents



Internet access followed by viewing the retrieved contents - processor activity for user+kernel



McAfee VirusScan (Full Scan)
Dell XPS400 – 2 processor cores



Processor Activity for Playing a Music CD

Computer = Dell Dimension 8200

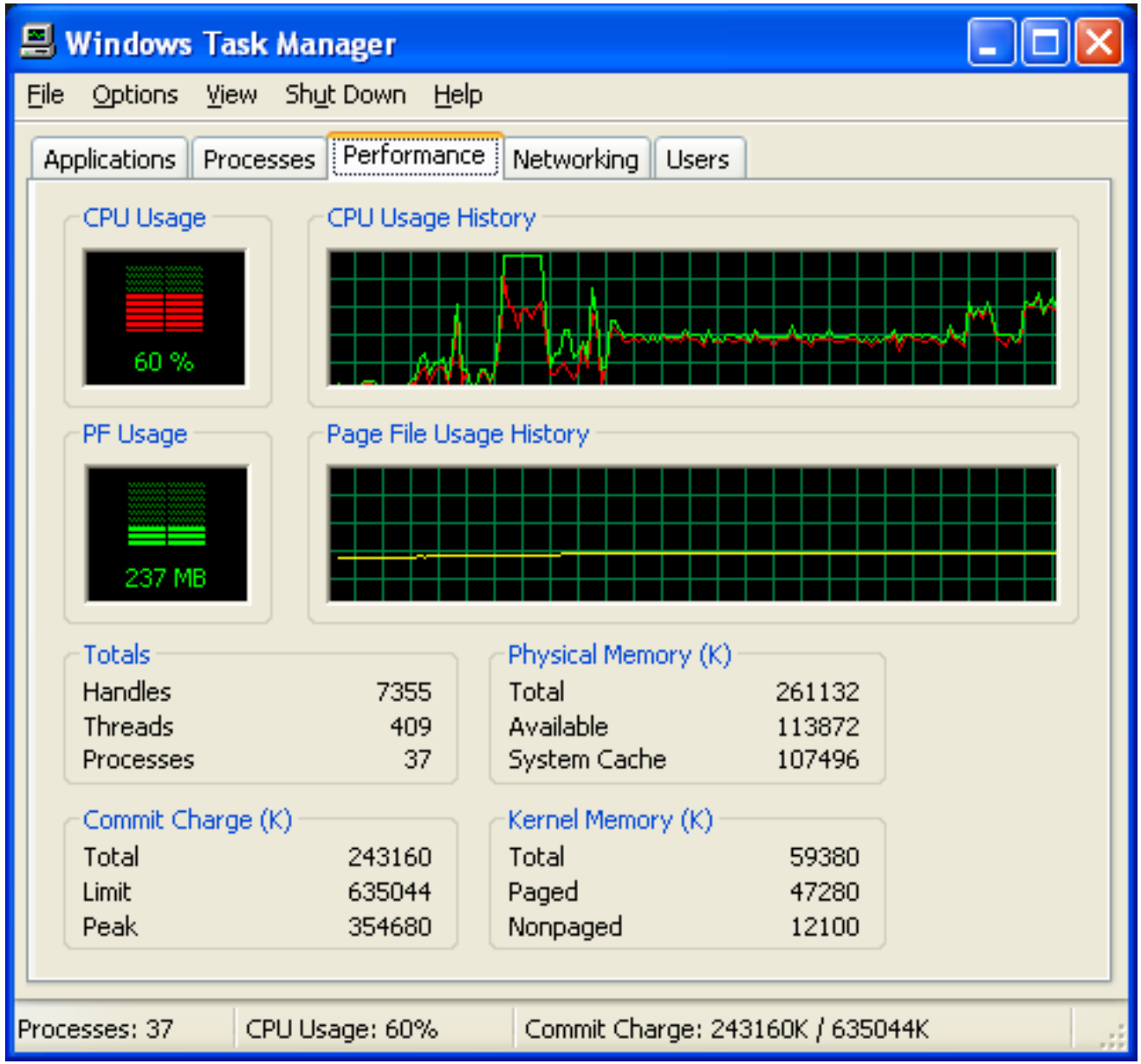
CPU = Pentium 4 @ 1.69 GHz

Memory = 256 MB of RAM

Average processor utilization = 50-60%

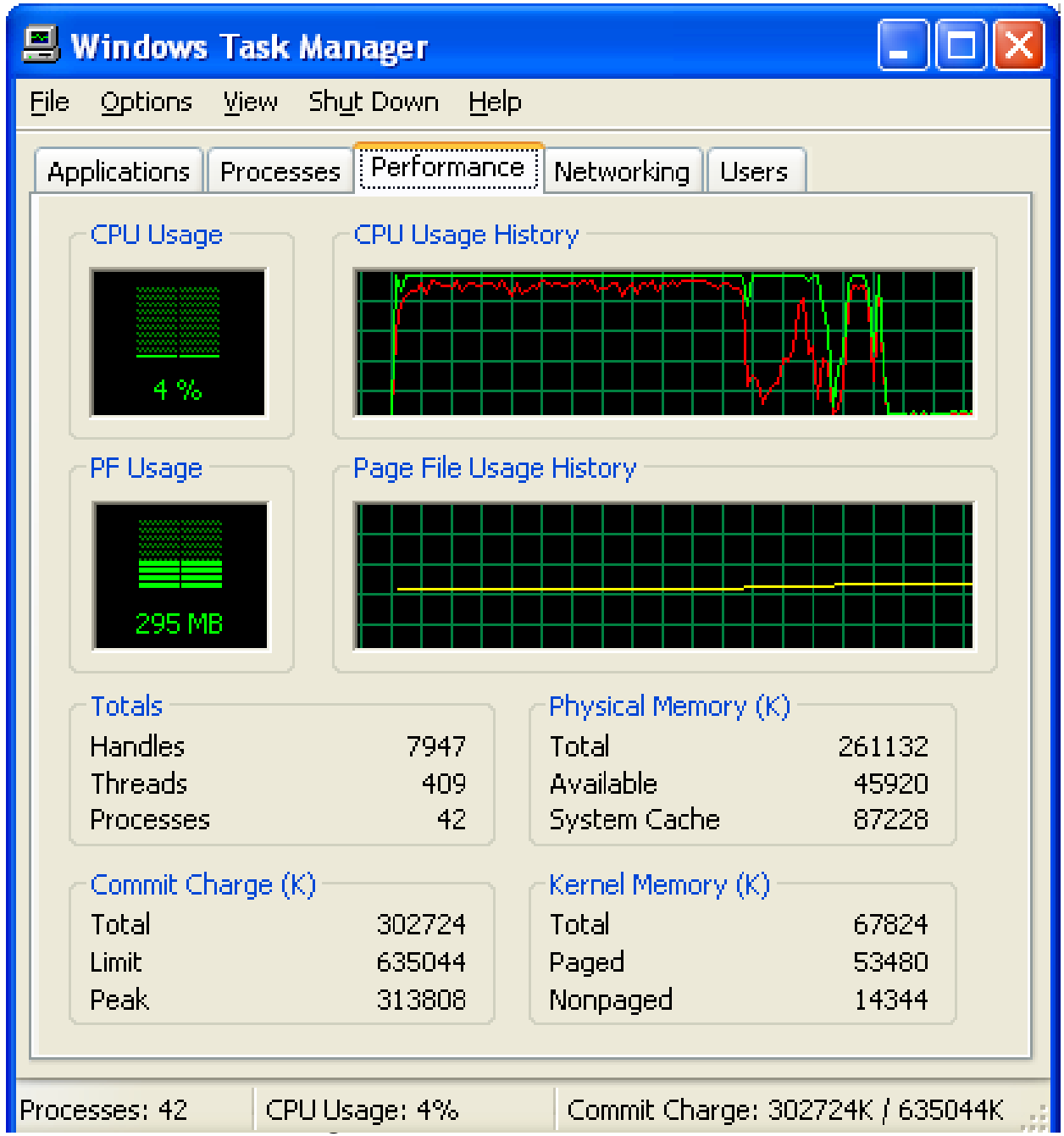
Initialization of this process takes more processor power.

Workload is mostly kernel activity.



Converting a Word Document to PDF file (Acrobat)

Dell Dimension 8200

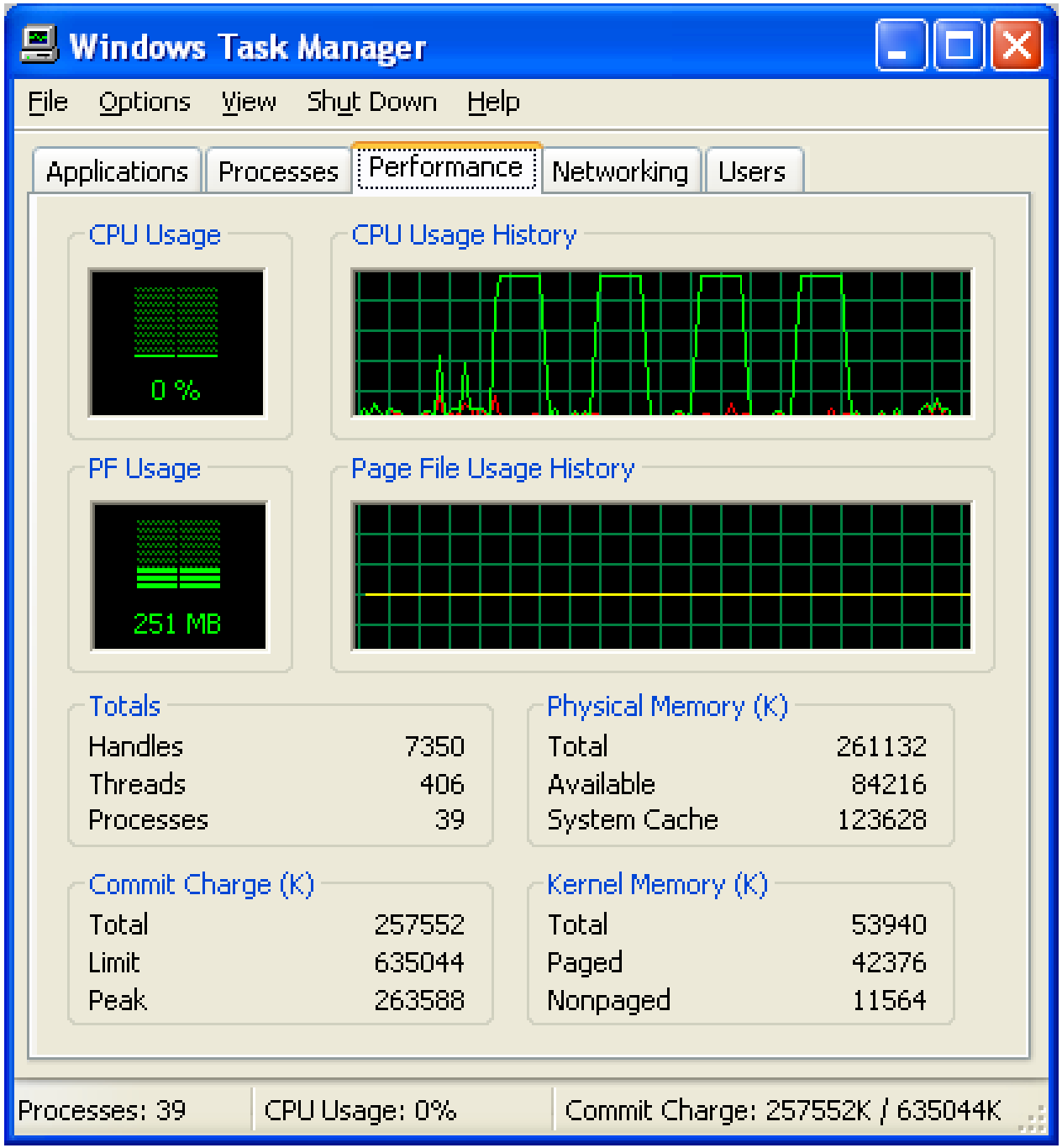


Jozo Dujmović

The Flip-Flop Program

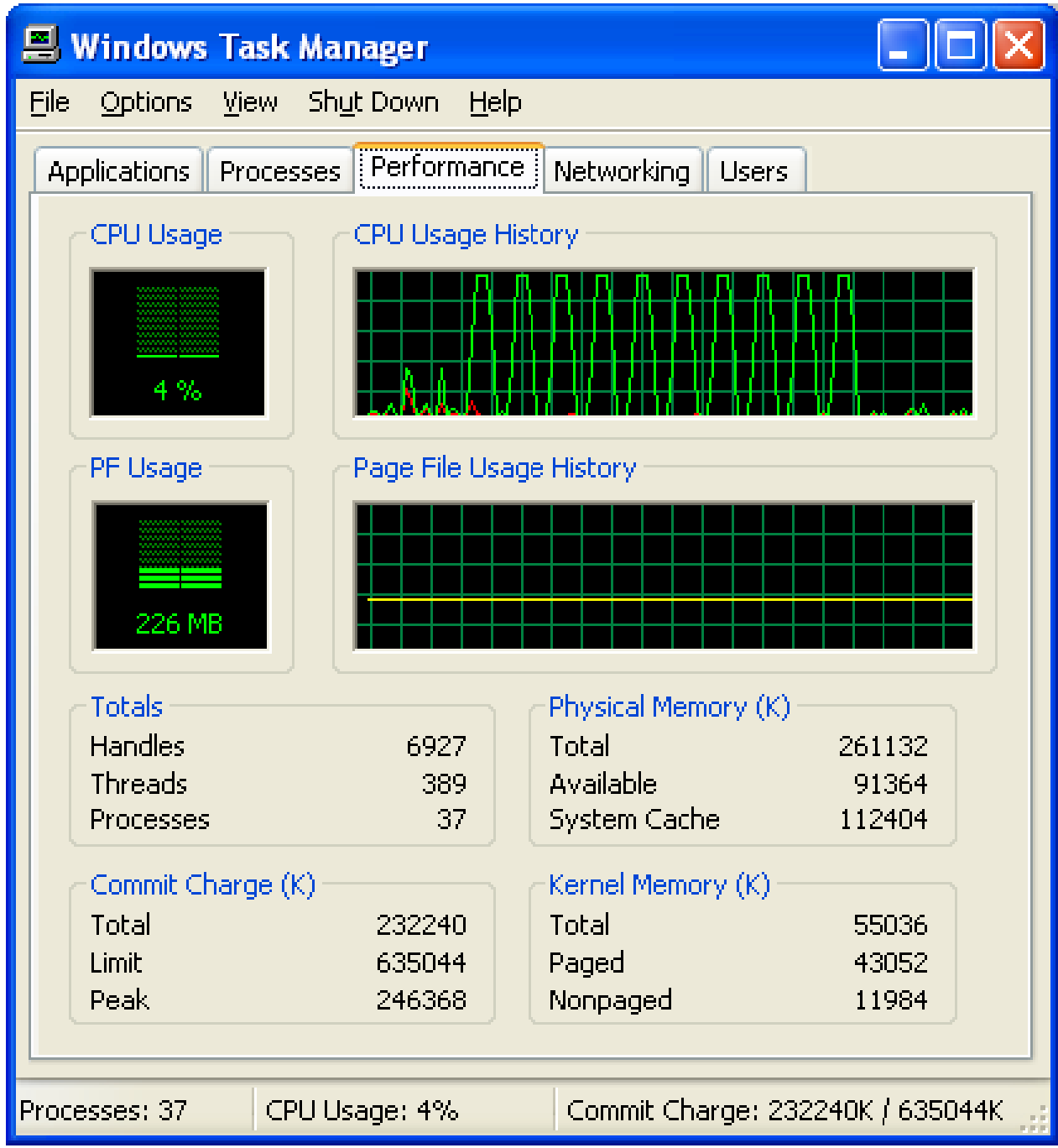
```
#include <iostream.h>
#include <time.h>              //  clock_t clock( void );
#include <Windows.h>           //  void Sleep(unsigned long MilliSeconds)

void Run(unsigned long Milliseconds)
{
    clock_t EndTick = clock_t(0.001*Milliseconds*CLOCKS_PER_SEC) + clock( );
    while(clock( ) < EndTick);
}

void main(void)
{
    unsigned long TotalTime      = 40000,
                  NumberOfCycles = 80,
                  ms = TotalTime/2/NumberOfCycles, i;
    for(i=0 ; i<NumberOfCycles ; i++)
   {
       Run(ms);
       Sleep(ms);
    }
}
```

Compiling and execution of 4 flip-flop cycles

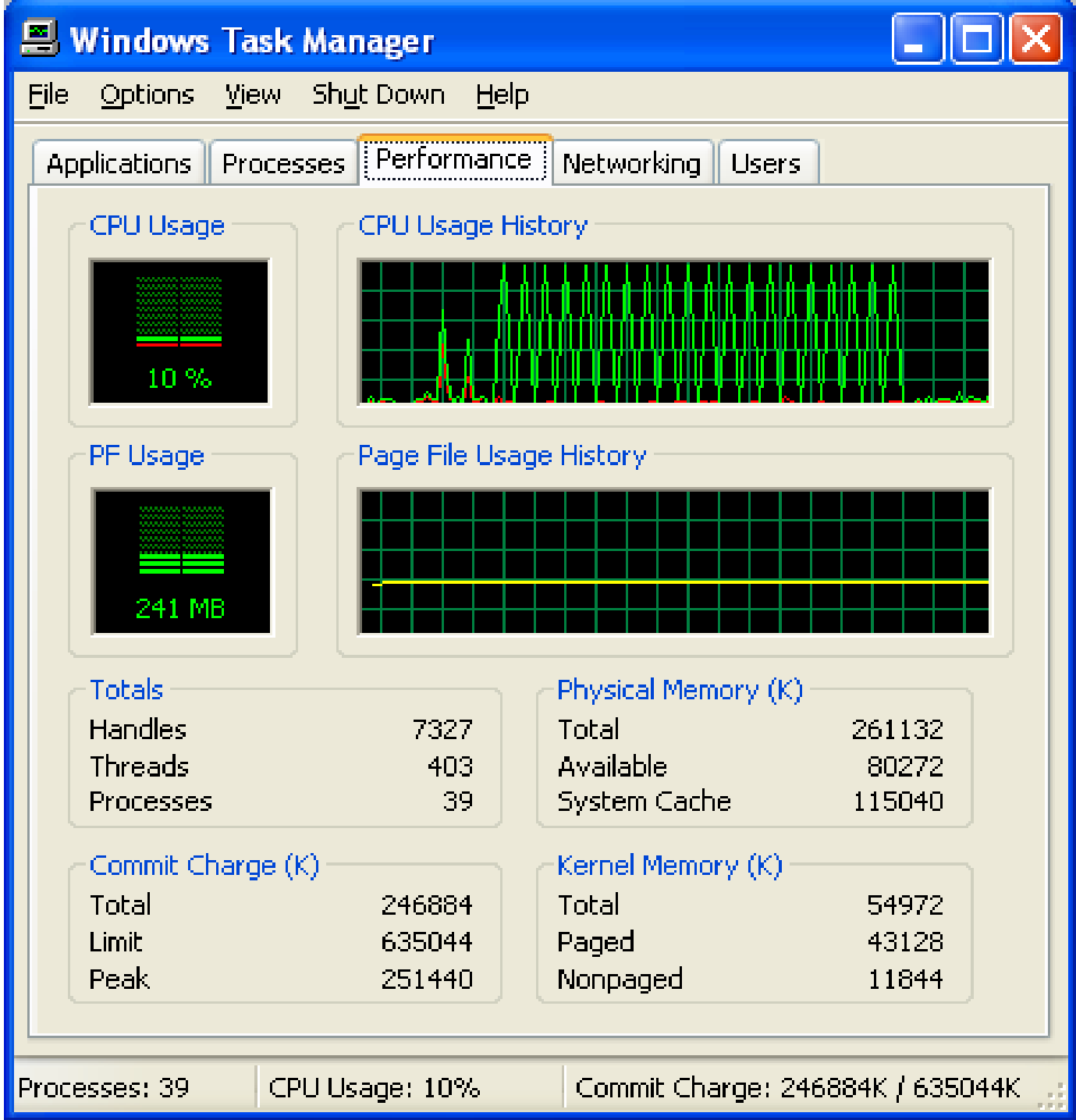


Jozo Dujmović

Compiling and execution of 10 flip-flop cycles

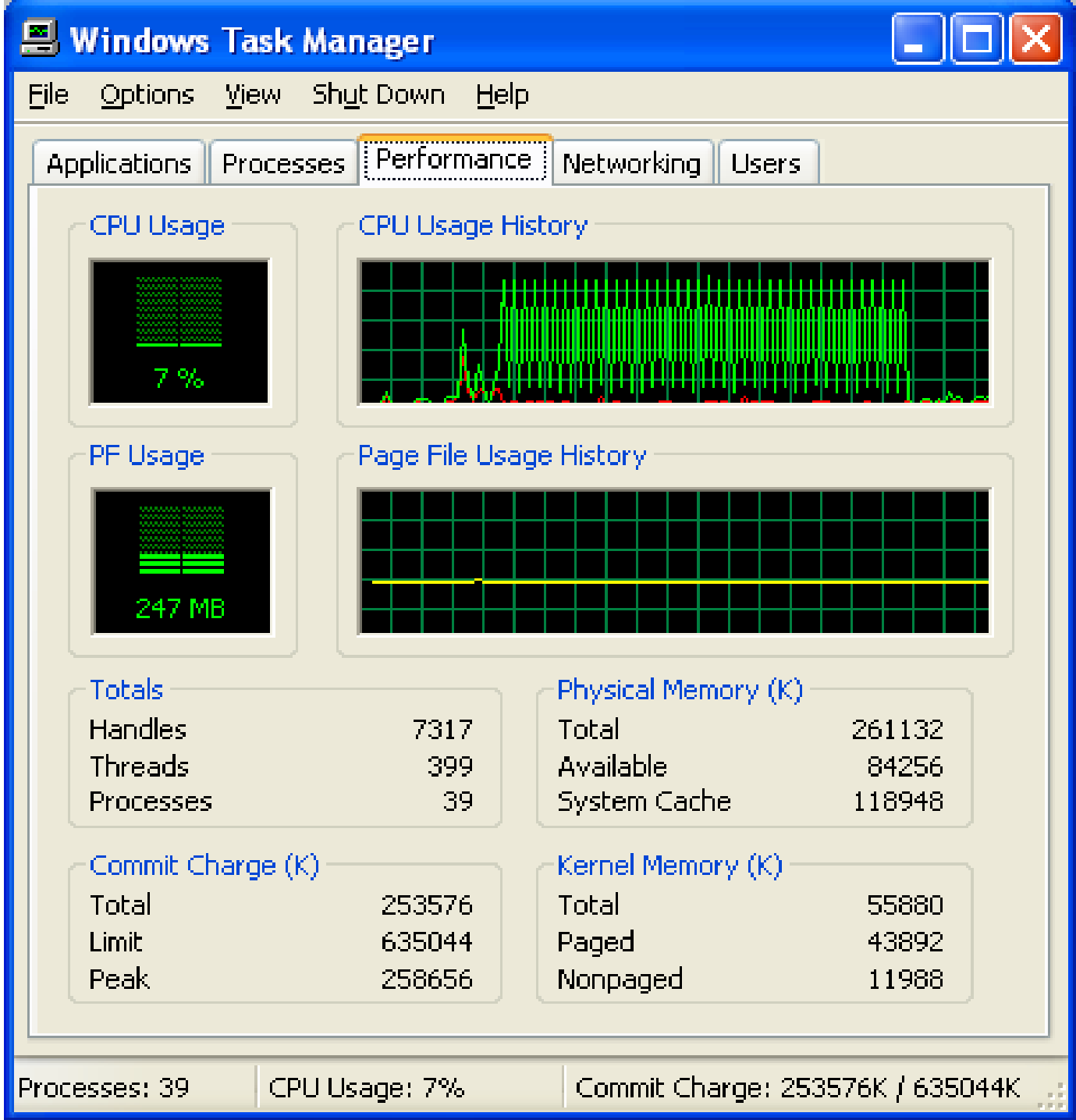


Compiling and execution of 20 flip-flop cycles



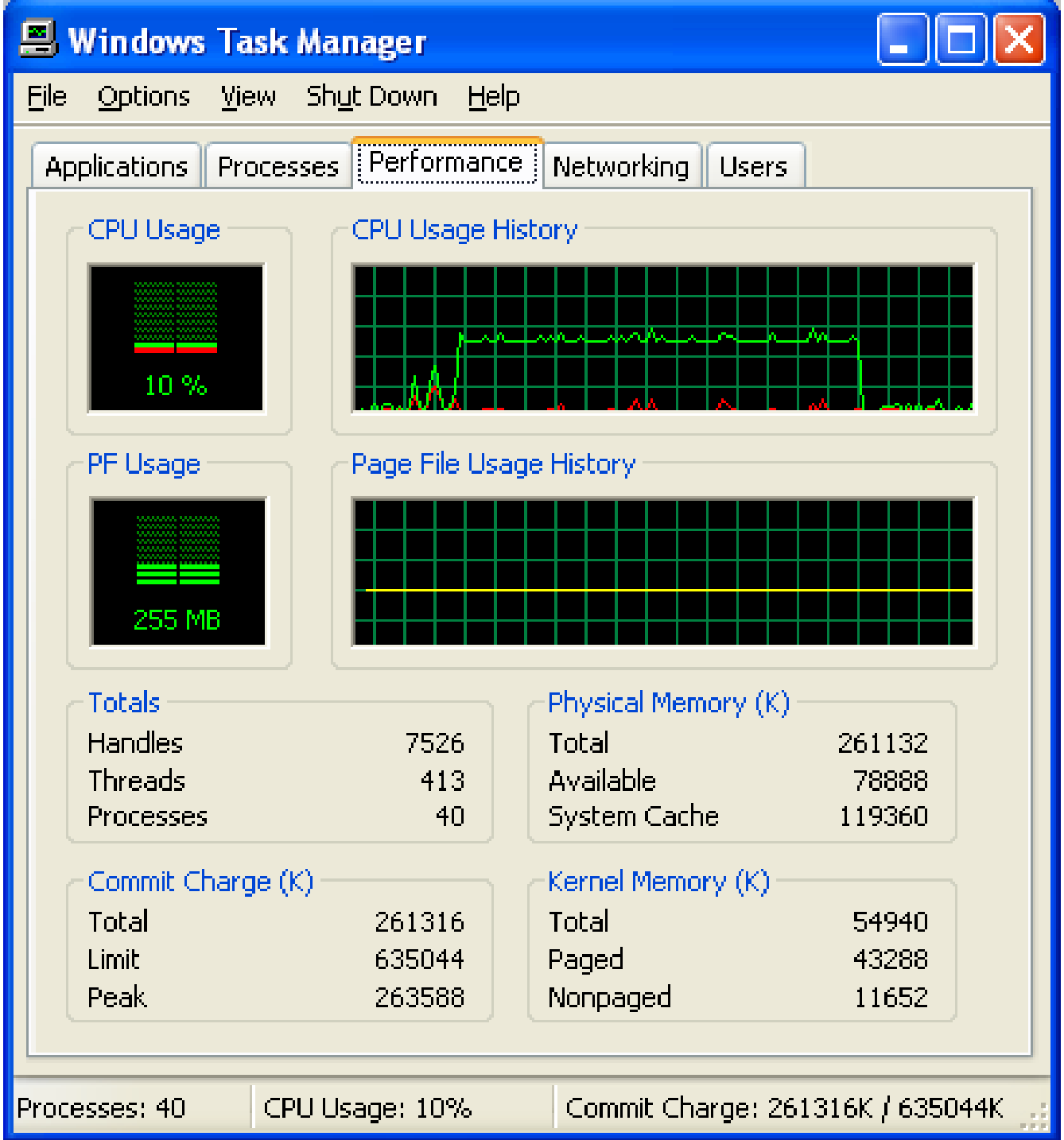
Jozo Dujmović

Compiling and execution of 40 flip-flop cycles



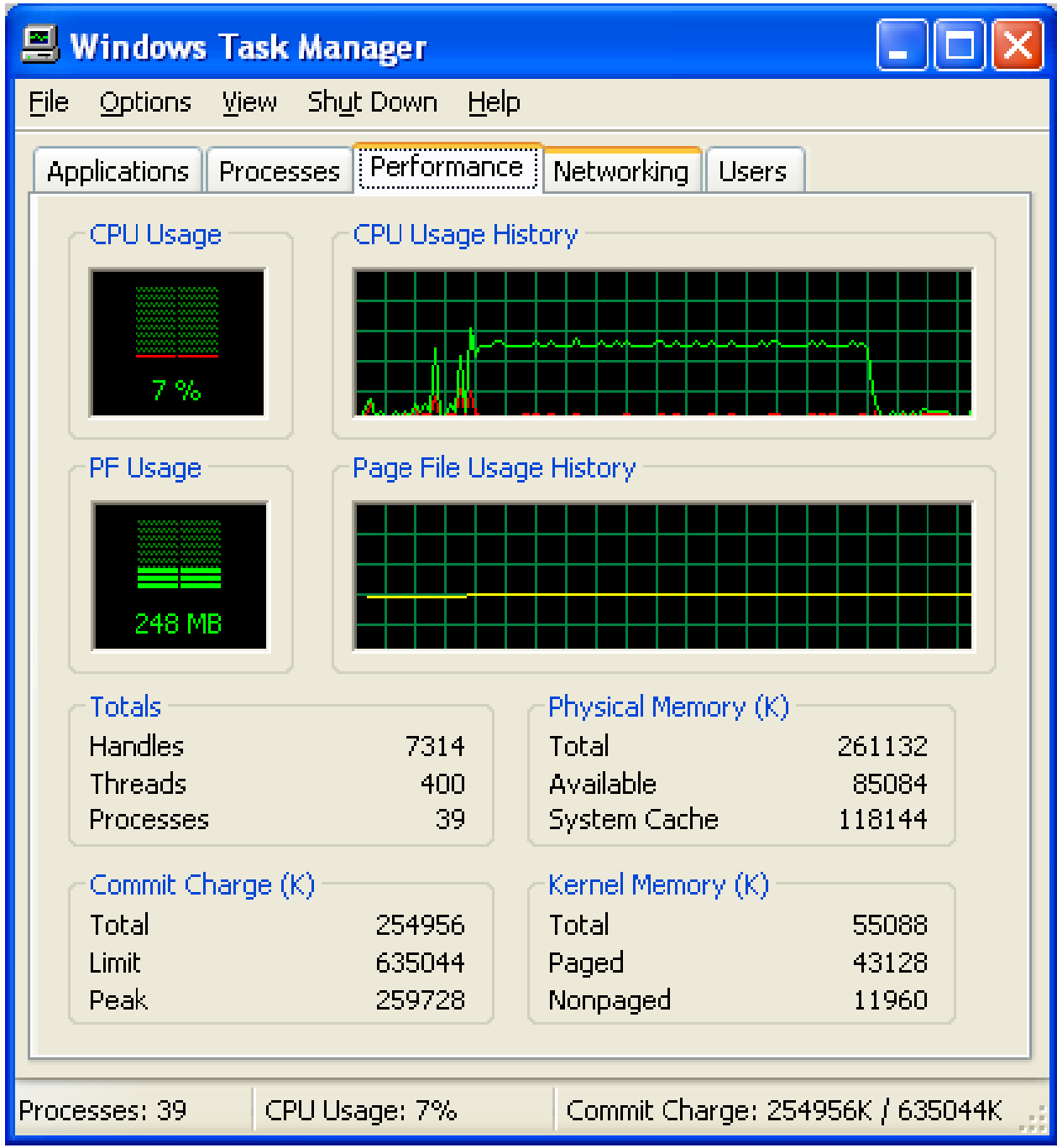
Jozo Dujmović

Compiling and execution of 80 flip-flop cycles



Jozo Dujmović

Compiling and execution of 160 flip-flop cycles



Measurement of processor utilization

- We use the modified FlopFlop program to alternate intervals of 4 seconds of processor activity followed by 4 seconds of idling.
- This pattern should produce total accumulated processor time that is exactly 50% of the total elapsed physical time.
- Processor utilization should be 50%

MODIFIED FlipFlop PROGRAM

```
#include <iostream.h>
#include <time.h>          //  clock_t clock( void );
#include <Windows.h>       //  void Sleep(unsigned long MilliSeconds)

double sec(void) {return double(clock())/double(CLOCKS_PER_SEC);}

void Run(unsigned long Millisec)
{
  clock_t EndTick = clock_t(0.001*Millisec*CLOCKS_PER_SEC)+clock( );
  while(clock( ) < EndTick);
}

void main(void)
{ unsigned long TotalTime      = 40000,    // Total time = 40 sec
                NumberOfCycles = 5,        // 8 seconds per cycle
                i, ms = TotalTime/2/NumberOfCycles;  // ms = 4 sec
  double CPUtime, RealTime;

  CPUtime  = sec();                  // Accumulated processor time
  RealTime = time(NULL);             // Physical time
  for(i=0 ; i<NumberOfCycles ; i++)
  {                                  // Each cycle contains 50% of
      Run(ms);                       // processor activity followed by
      Sleep(ms);                     // 50% of idling with intention
  }                                  // to cause processor utilization
  CPUtime = sec() - CPUtime;         // of exactly 50%
  RealTime = time(NULL) - RealTime;

  cout << "\nCPU time   = " << CPUtime  << " sec"
            << "\nRealTime   = " << RealTime << " sec"
            << "\nProcessor utilization = "
            << 100.* CPUtime/RealTime << " %\n\n";
}
```

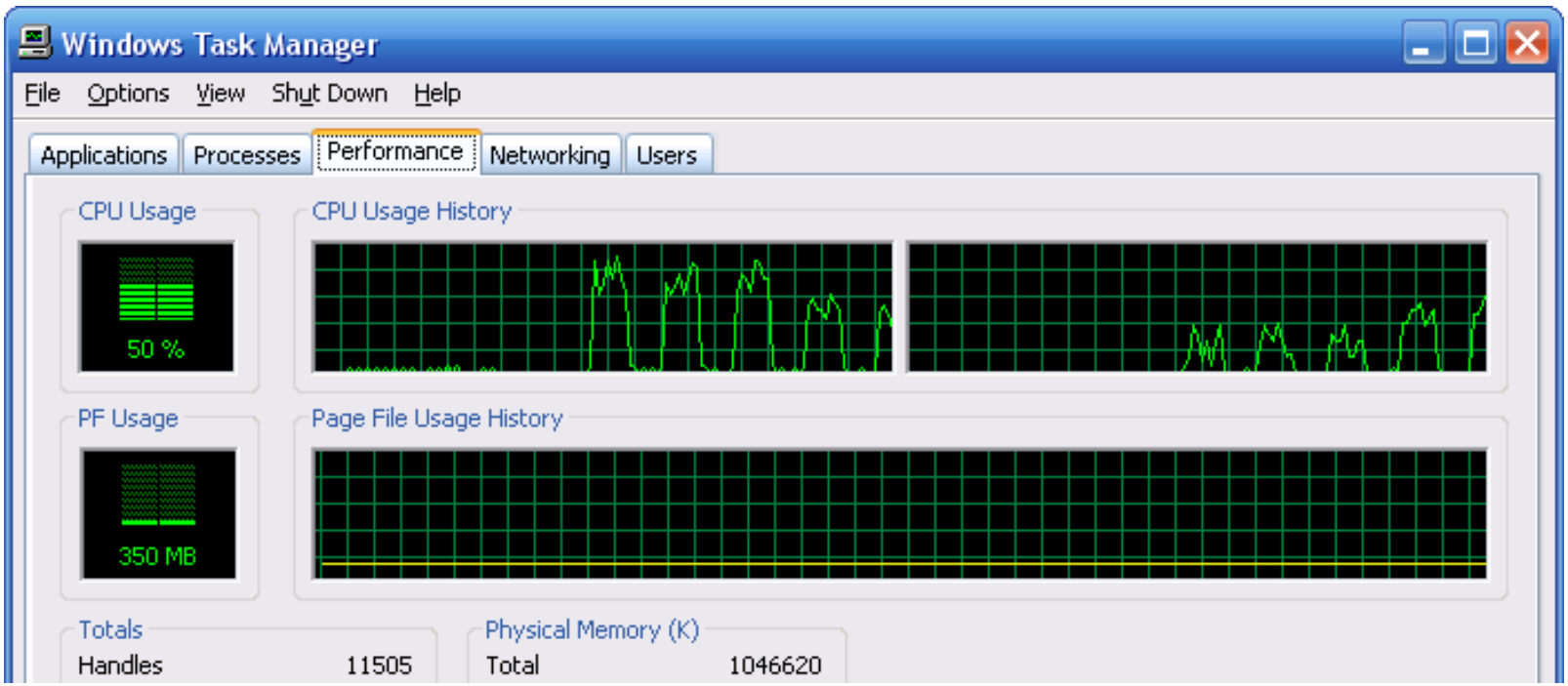
Modified FlipFlop program for measurement of processor time and real time during the execution of FlipFlop workload that uses processor 50% of time

Under Windows XP this program generates wrong results caused by clock function that returns physical time instead of CPU time.

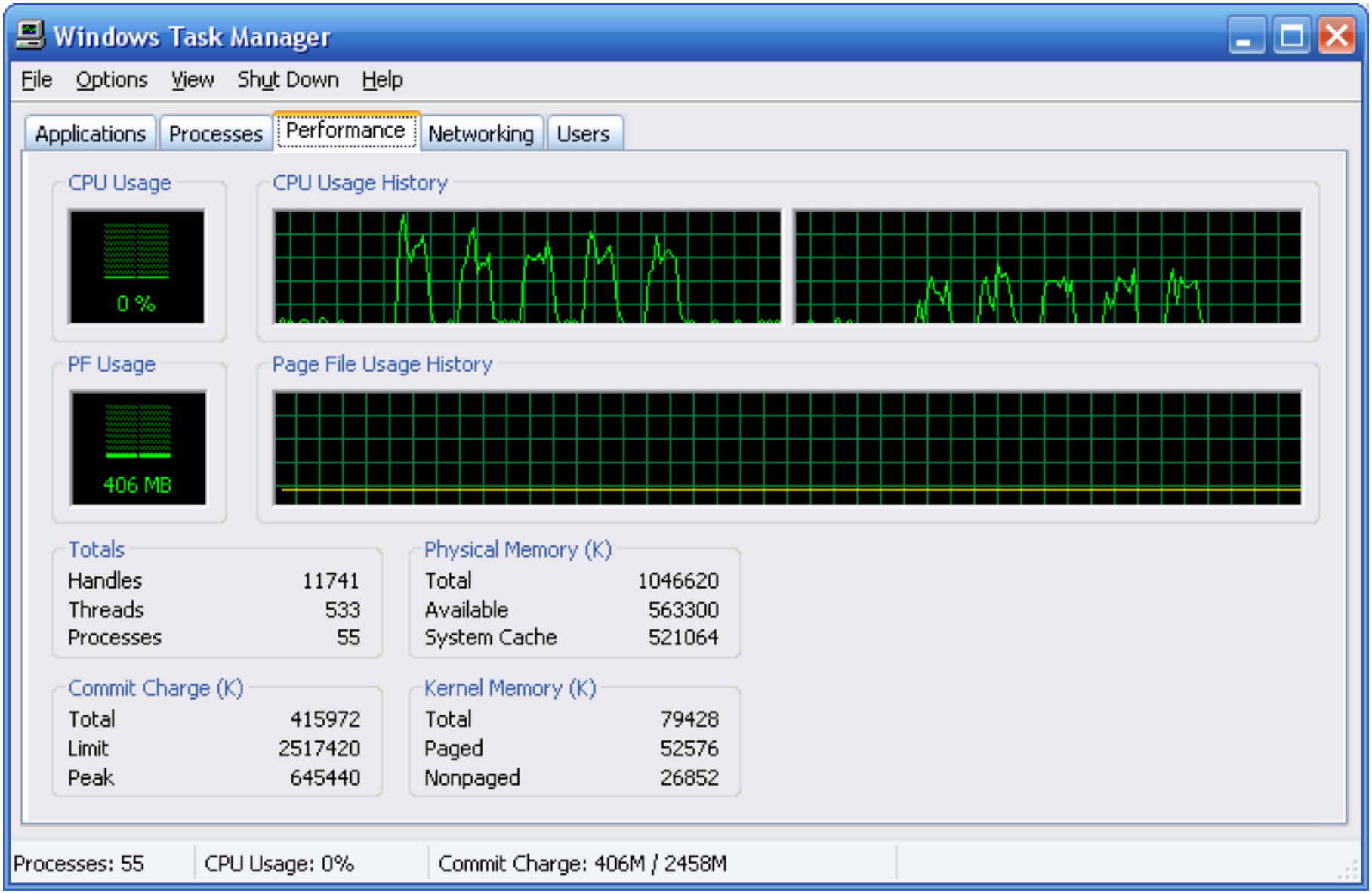
Wrong results generated by the modified FlipFlop program

```
CPU time   = 40 sec
RealTime   = 40 sec
Processor utilization = 100 %
```

?



Task manager clearly shows 5 FlipFlop cycles that cause processor utilization = 50%



Processor time vs. real time

- Processor time consists of the sum of all processor quanta assigned to a specific process
- Real time is the physical elapsed time
- Generally: Real time = processor busy time + processor idle time