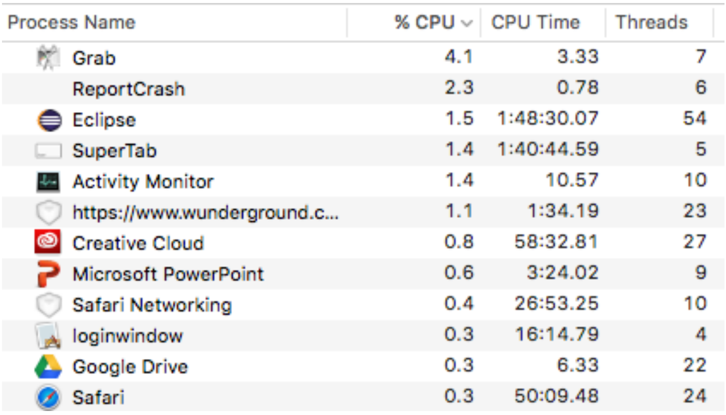
# Cores and processing units, processes and threads

The image to the right is part of a table of the *Activity monitor* on a Macintosh laptop. It was created in 2015. The table shows a few of the over-100 processes that were being concurrently executed. Some are applications that the user opened, like Eclipse, PowerPoint, Safari, and the Activity Monitor itself. Many of them are parts of the operating system —keeping it functioning properly and reacting to events.

The second and third columns give the percent of the CPU (*Central Processor Unit*) time and the CPU time itself that the processor took in the past period of execution for which statistics were kept.



| Process Name | % CPU ⌄ | CPU Time | Threads |
|---|---|---|---|
| Grab | 4.1 | 3.33 | 7 |
| ReportCrash | 2.3 | 0.78 | 6 |
| Eclipse | 1.5 | 1:48:30.07 | 54 |
| SuperTab | 1.4 | 1:40:44.59 | 5 |
| Activity Monitor | 1.4 | 10.57 | 10 |
| https://www.wunderground.c... | 1.1 | 1:34.19 | 23 |
| Creative Cloud | 0.8 | 58:32.81 | 27 |
| Microsoft PowerPoint | 0.6 | 3:24.02 | 9 |
| Safari Networking | 0.4 | 26:53.25 | 10 |
| loginwindow | 0.3 | 16:14.79 | 4 |
| Google Drive | 0.3 | 6.33 | 22 |
| Safari | 0.3 | 50:09.48 | 24 |

A processing unit is part of the computer that actually executes a sequence of machine instructions. CPU is a term from the past, when computers had *one* processing unit, called the *Central* processing unit. If there is only one unit, only one process can be executed at a time. The operating system cycles through the processes, giving each a "time slice" of execution[1]. Different processes have different priorities, and the higher priority threads get more time. The operating system has a *scheduler* —itself a process— which is able to interrupt a process that is currently running and determine which process should run next. Depending on the computer and its speed, the time slice could be around 100 milliseconds.

Nowadays, the processor on your computer, a chip, generally has several independent processing units called *cores*, each of which is able to execute a (different) sequence of instructions. The 15-inch Macbook Pro on which this little essay is being typed, built in 2014, has one four-core processor, so the operating system has to manage 4 processing units in giving processes time slices. You can now buy Intel processors with 10 or more cores.

Look at the last column of the table. Eclipse has 54 *threads*. Safari has 24. A *thread* is a sequence of instructions that is being executed or is waiting to be executed. Most applications start and stop threads at various times. For example, suppose a Java GUI program that you wrote is running in Eclipse. When you click a button of the GUI, a new thread of execution is created to process the click. The new thread will contain the code that calls the method that processes the click. This thread runs in parallel with any other threads that your Java program has created.

Eclipse has many more threads going. Some handle the editing that you are doing, others are involved in syntax-checking and compiling Java classes as you edit them, others in updating the Eclipse window, and so on.

Java has features that allow *you*, the programmer, to create and destroy threads of execution. You can use those features to write your own parallel program. For example, you could write a program that simulates a home heating system, with different threads for the furnace, the temperature inside the house, the temperature outside the house, and the thermostat.

## Exercise

Start your Activity Monitor (it's called the *Resource Monitor* in Windows) and watch the table for a minute, see how it changes. Start Eclipse, and move your windows around on your monitor so that you can see windows of both Eclipse and Activity Monitor. Add a static method with an infinite loop to some class, and call that method. See how a process called java appears, how many threads it has, and how much CPU time it takes. Do the same thing with DrJava. This exercise will reinforce what you read and make it more meaningful.

[1] A paper titled *Time* Slices: What *Is the Duration of a Percept*?, written by Herzog, Kammer, and Scharnowski and published on 12 April 2016, suggests that your brain time slices also, with slices being as much as 400 milliseconds. See https://doi.org/10.1371/journal.pbio.1002433.