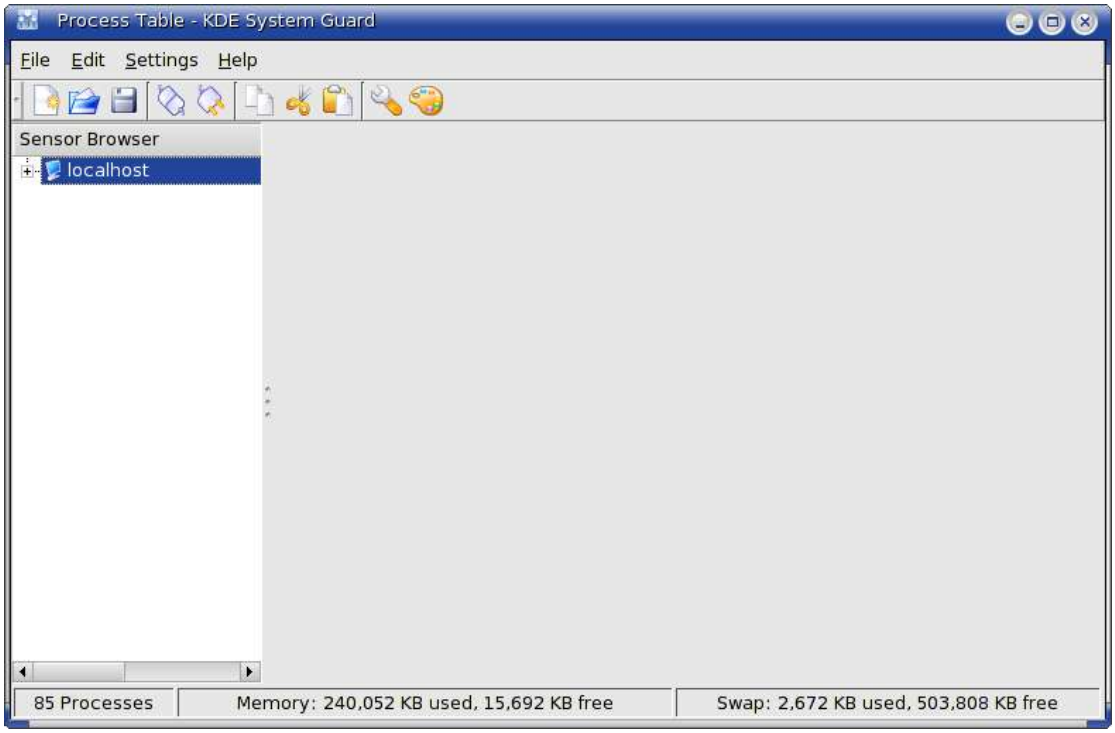
System Monitoring with KSysGuard

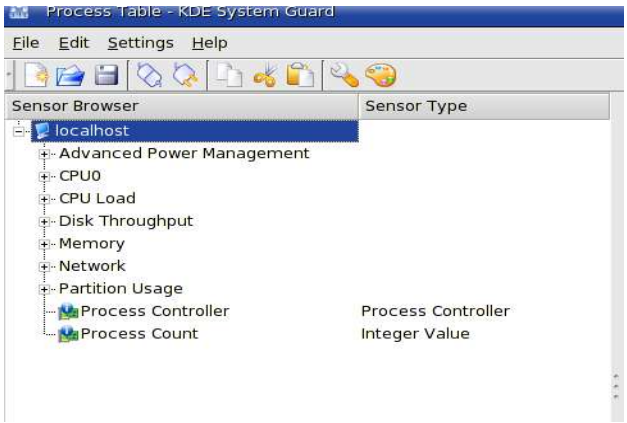
In the days of a Linux administrator, it is almost invariable that he/she will need to monitor the system(s) for any bottlenecks. There are several programs available for this, several daemons available, but we will cover a fairly powerful and flexible application called KsysGuard.

KsysGuard is accessible at least three ways: typing 'ksysguard' on the command line, clicking K-menu > System > Monitoring > KsysGuard, or CTRL-ESC (this usually brings up only the process list). Windows users will see this application as a morph between Task Manager and Performance Logs and Alerts. Notice the information already shown at the bottom of the window.

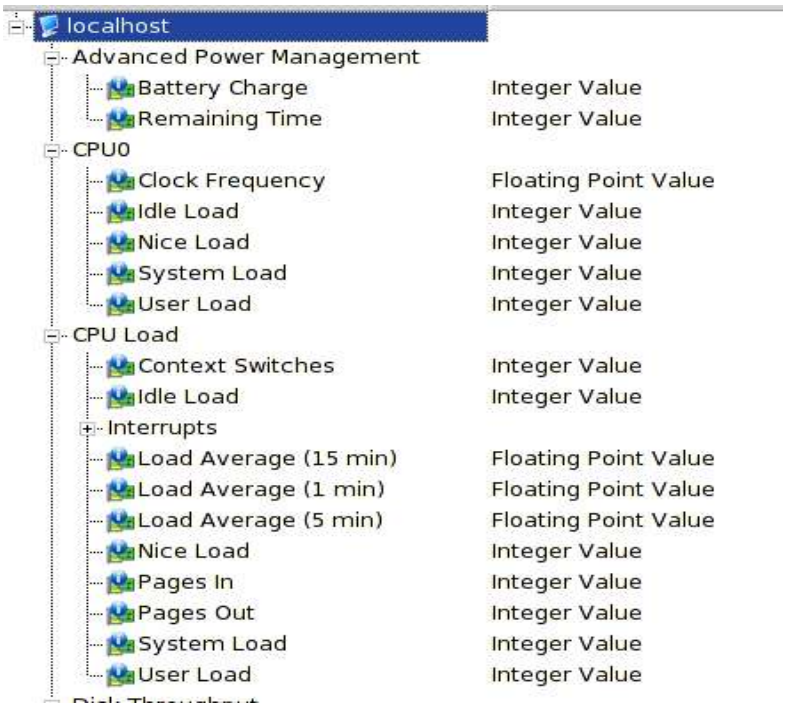


Be sure to explore the buttons along the top: New, Open, Save, Connect Host, Disconnect Host, Copy, Cut, Paste, Worksheet Properties, and Configure Style. We'll be using a few of these during this discussion, but you'll want to be at least familiar with them once you are ready to feel out this program.

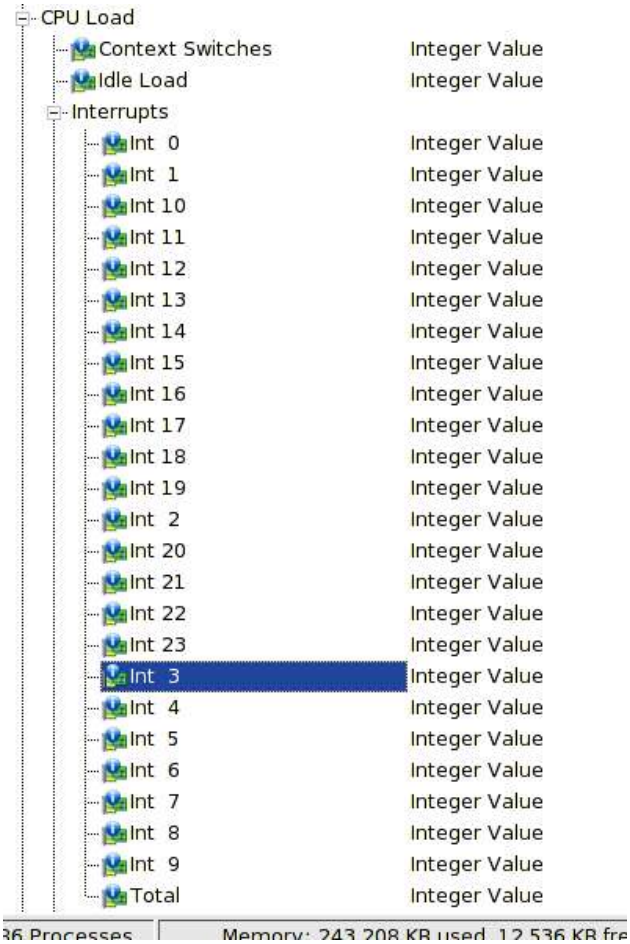
Under “Sensor Browser”, you'll find all the sensors you can monitor. Think of these sensors as the answer to questions like, “What is the average load on my CPU over the past minute?” Here are the major categories:



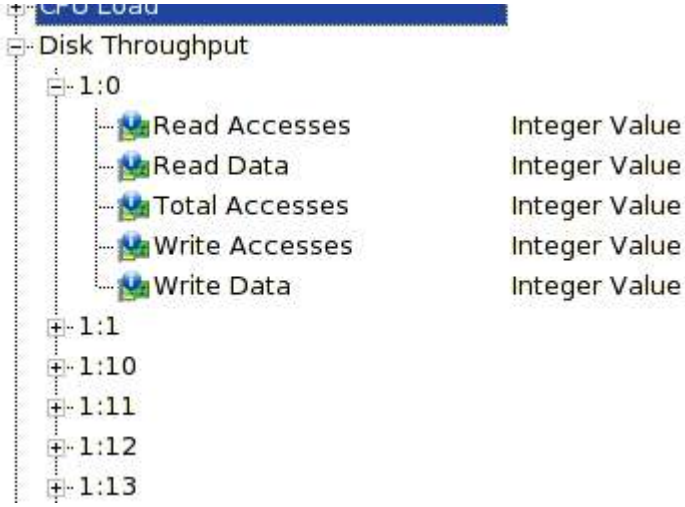
Here are the sensors under Advanced Power Management, CPU0 (the only CPU on this computer), and CPU Load:



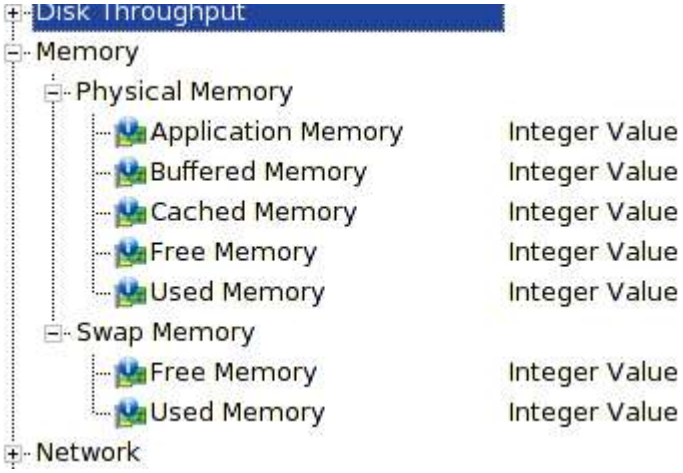
Under CPU Load, here are the sensors under Interrupts:



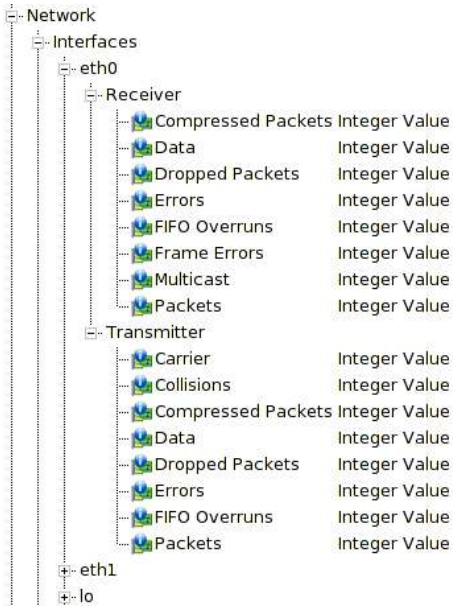
The next major category down is “Disk Throughput”, and here are the sensors available:



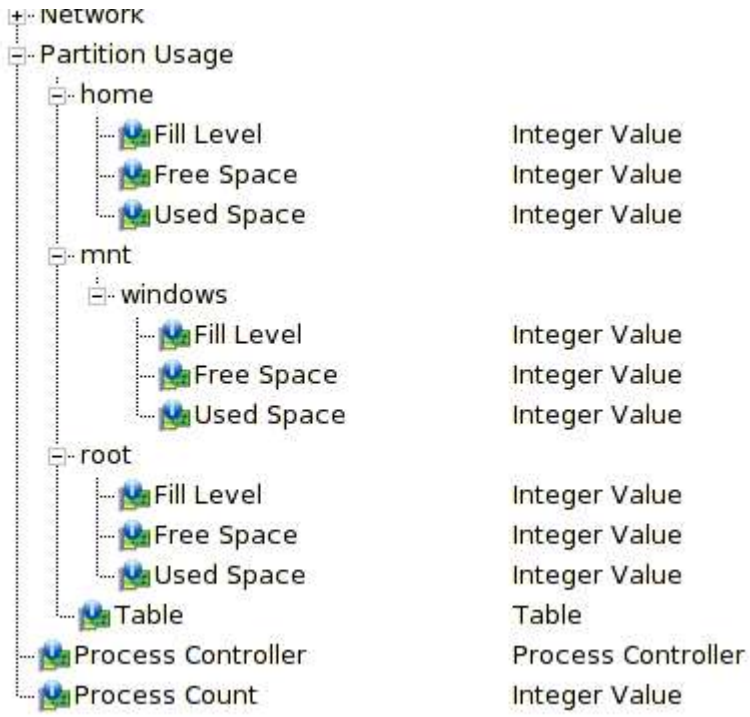
Under Memory, both Physical and Swap memory:



Now, under Network, you'll find your interfaces and their corresponding sensors:



The next major category is Partition Usage. Keep in mind that you can set an alert for Free Space, and you'll begin to realize a bit of the potential this could have for monitoring servers.

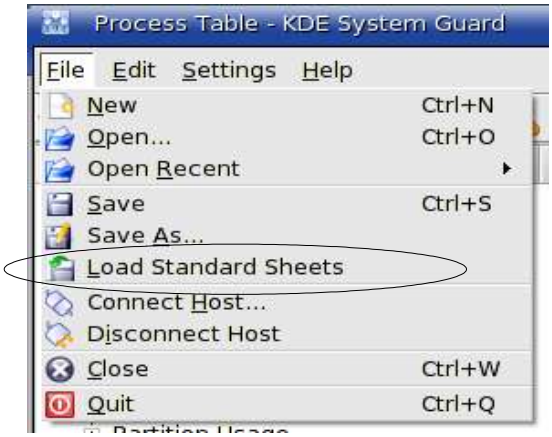


At the bottom of this, you'll see Process Controller and Process Count. Process Controller gives a single window effectively combining something like “top” and “kill”. Again, keep in mind this can be used on remote systems, and again you'll start seeing the potential for the Linux admin.

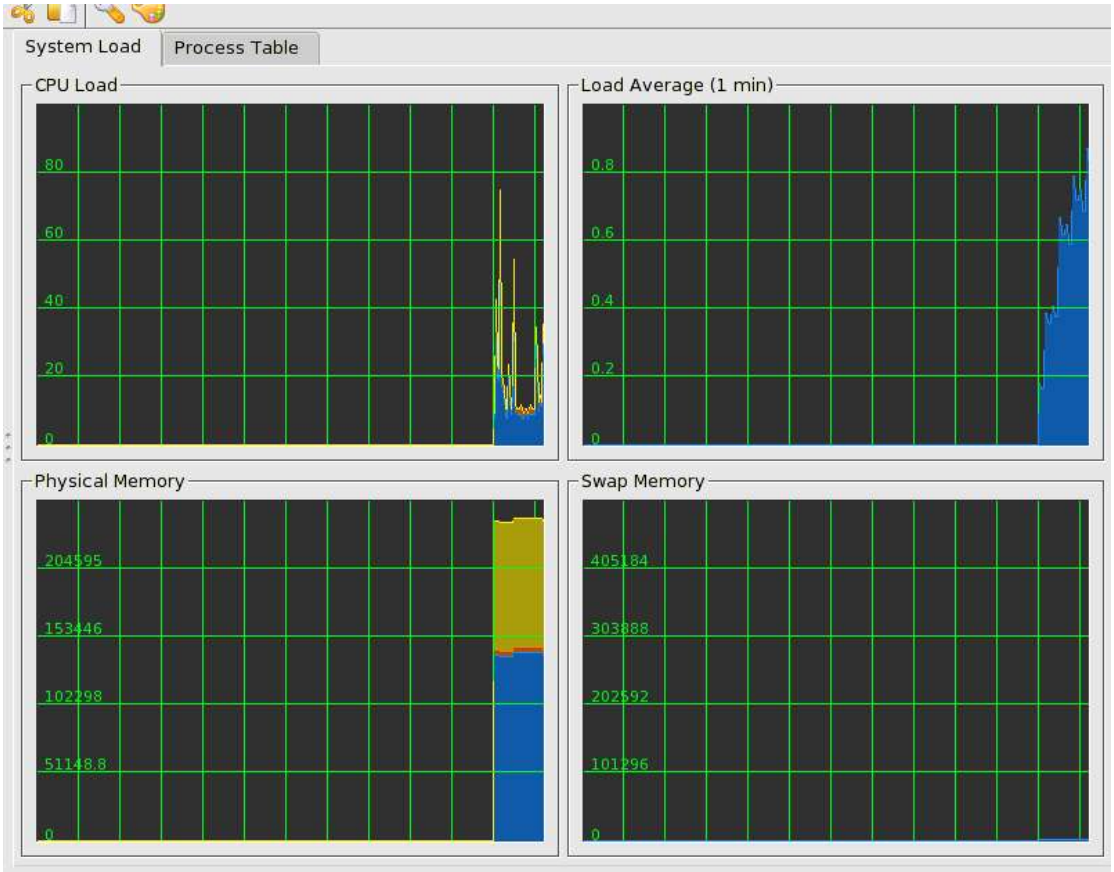
In order to being using the sensors, we need to start a worksheet, either by clicking File > New, or by clicking the button New.

Label the worksheet something meaningful. If you plan on just having CPU sensors, label it “CPU”. If you plan on having the process list, label it “Process List”.

At this point, instead of making a whole new worksheet, we are going to load in the standard worksheets and explore them in more detail. This will give us a starting point by giving us two worksheets already, “System Load” and “Process Table”.

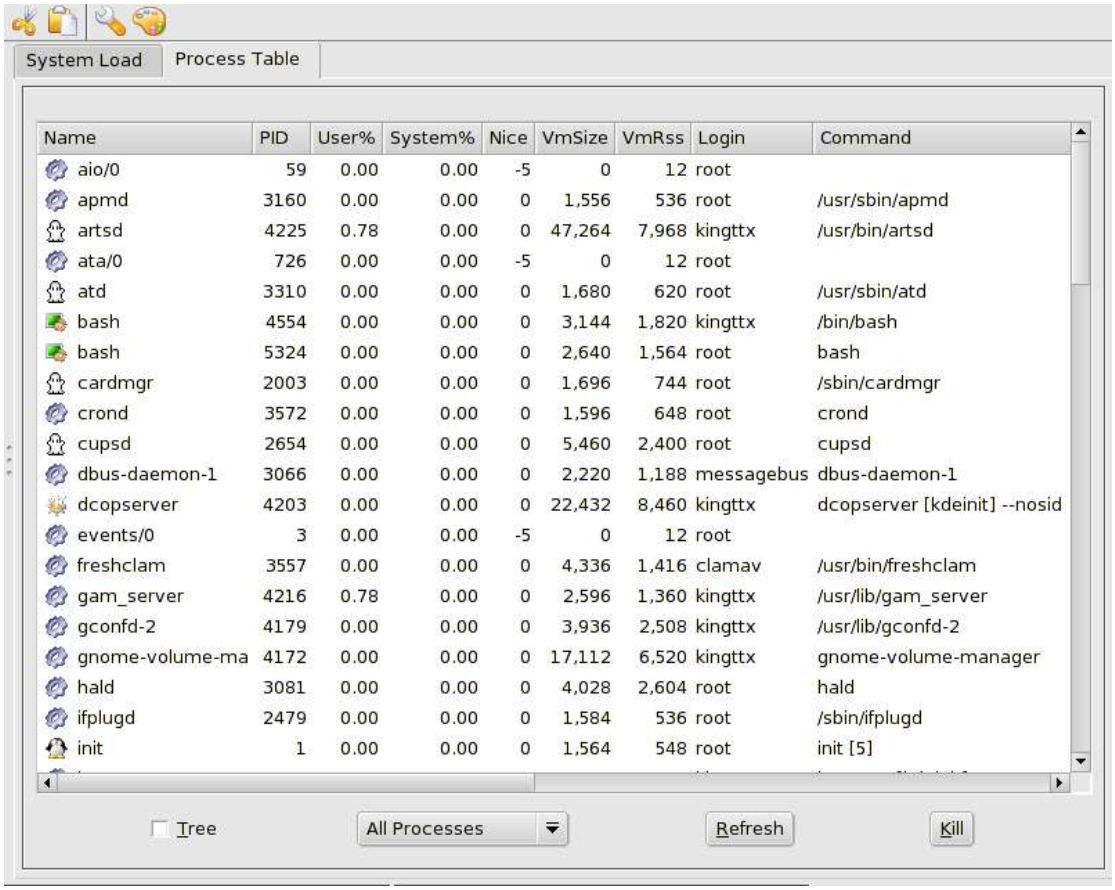


Here is the first worksheet:

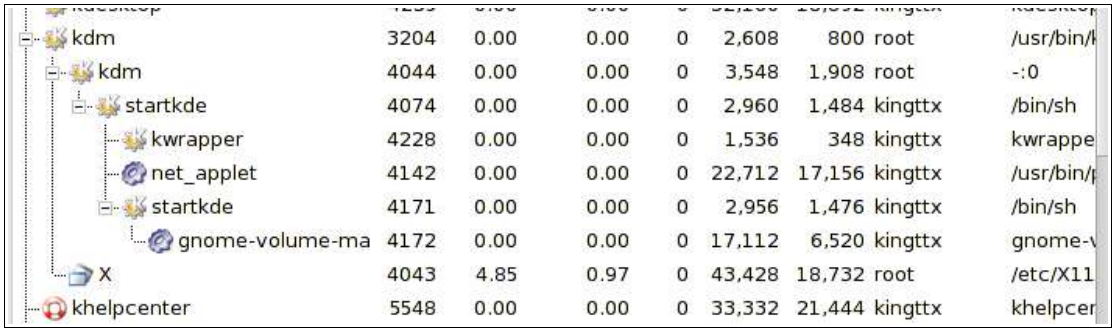


CPU Load consists of CPU Nice Load, CPU System Load, and CPU User Load.
Load Average (1 min) is fairly self explanatory.
Physical Memory consists of cached physical memory, buffered physical memory, and application physical memory.
Swap Memory is a single sensor showing used swap memory.
These particular graphs are cumulative; the sensors in a graph are added together for the entire height, and they are broken up using the color legend. We'll come back to these graphs in a little bit.

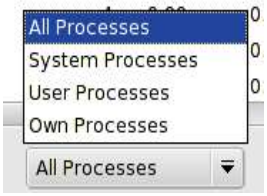
The next worksheet, Process Table, gives a list of current processes, similar to the ouput given by "top" on the command line.



Notice the difference after clicking the "Tree" checkbox:



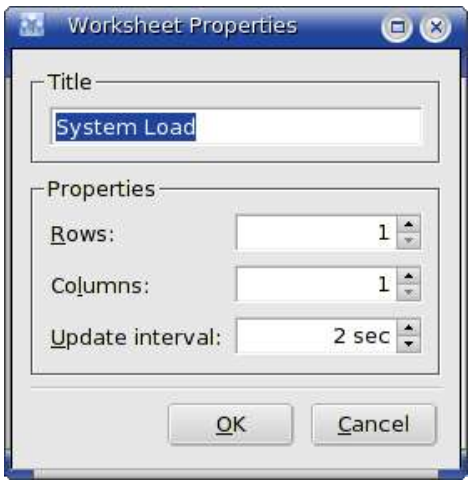
Next to "Tree", notice the drop down box. Here are all the filtering options:



"Refresh" will re-list the processes if needed, and "Kill" will kill a highlighted process.

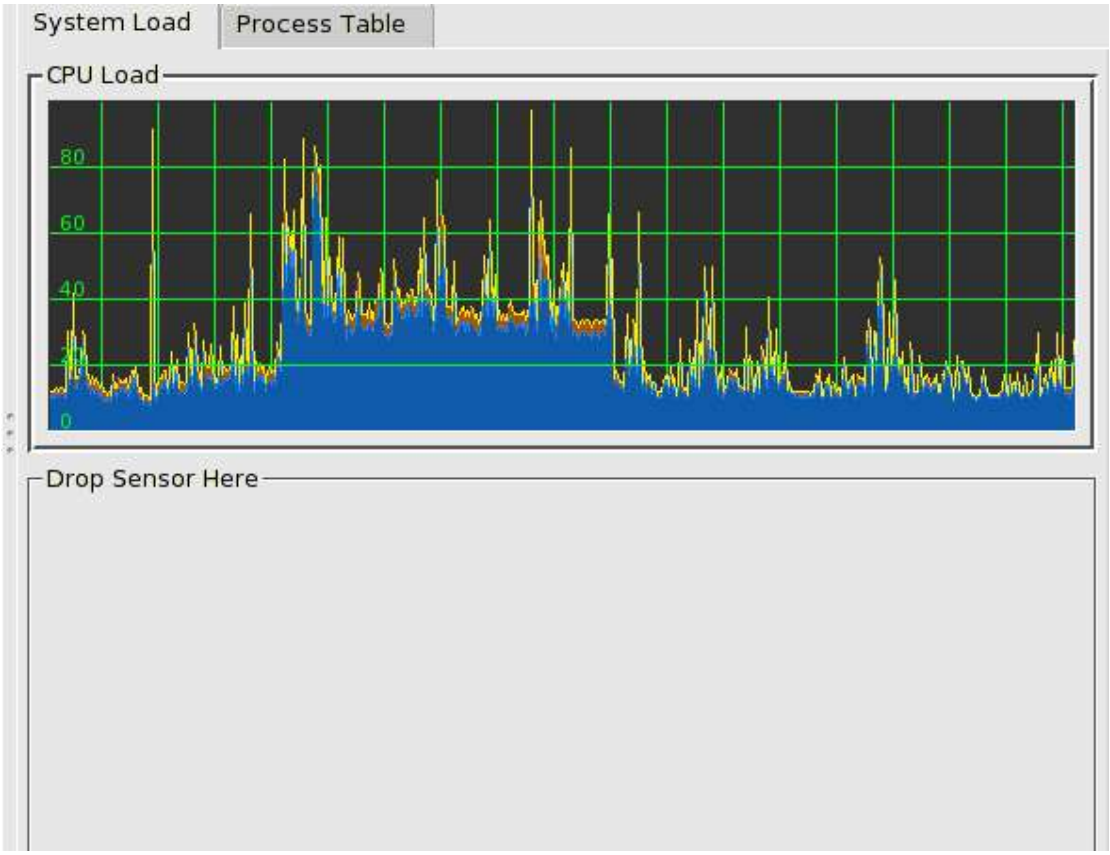
Let's return to the graph and show how to change properties of both a worksheet and a graph.

The Worksheet Properties can be accessed in at least two ways, either by clicking the properties button  or by clicking Edit > Worksheet Properties. This brings up the following dialog box. Notice I've already changed the amount of rows and cells to 1x1.

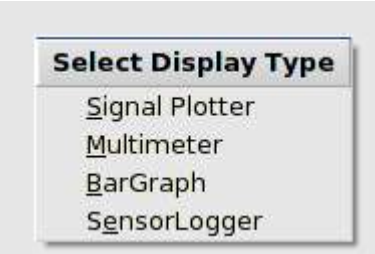


Note this important detail: The more graphs/sensors you have, and the smaller the update interval, the more load on your CPU to process the information and redraw your graphs or note your logs. Weigh this against your needs.

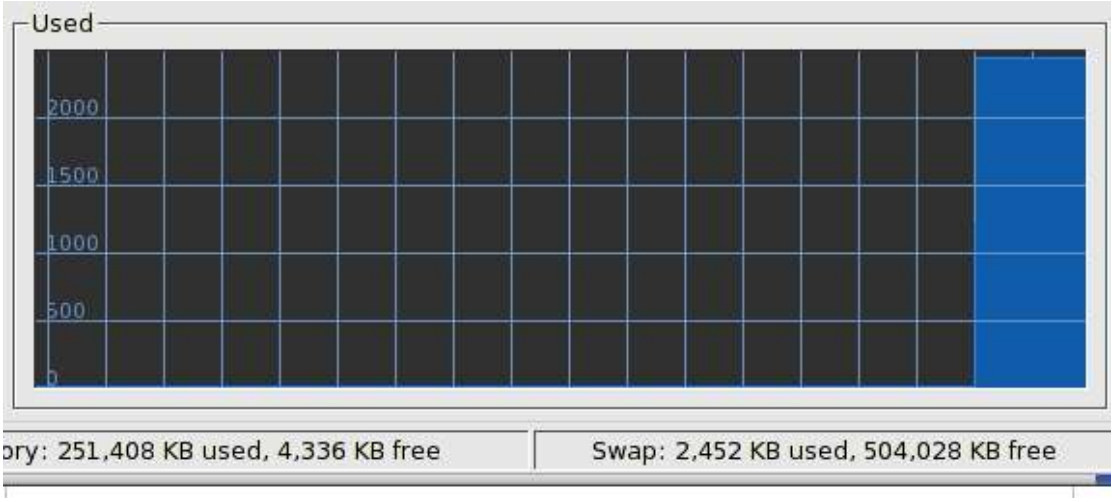
Let's say I wanted to add another graph. I'd open the properties and add a row or column and end up with something like this (1 column, 2 rows):



Let's drag and drop Swap Memory > Used Memory. Notice the popup once the mouse button is released:

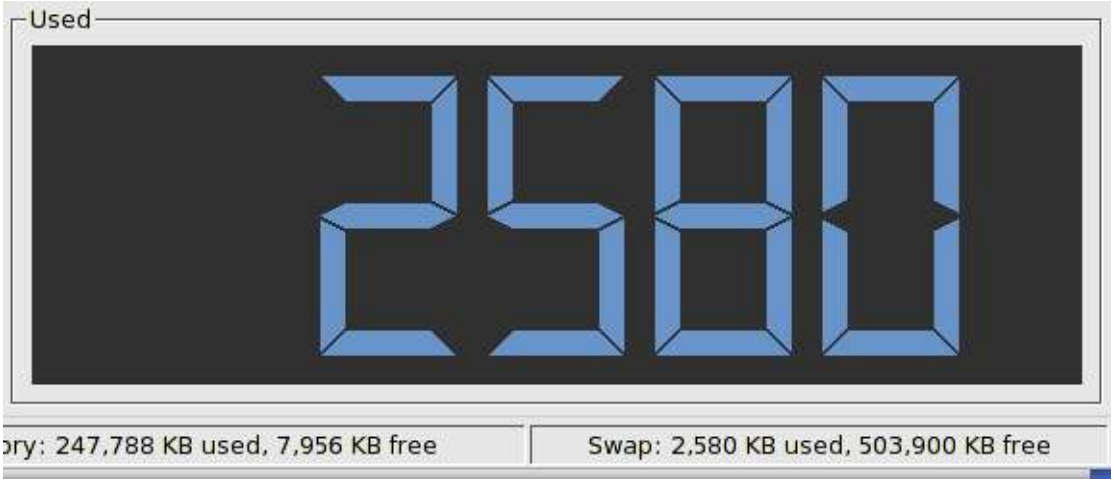


Here is the Signal Plotter:



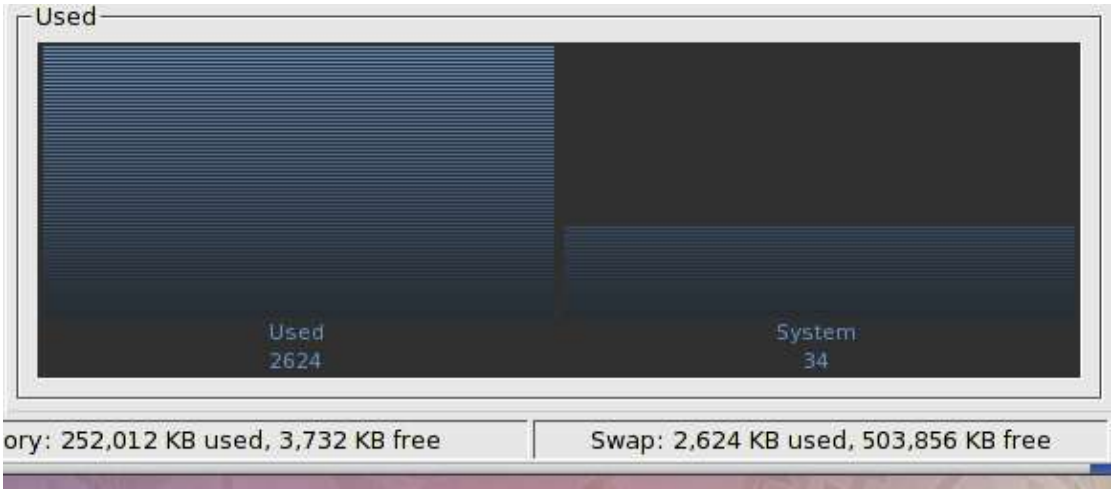
You can compare the numbers against the graph in this case since we picked a graph that KsysGuard normally notes in the status anyway.

Here is the same information as a Multimeter:

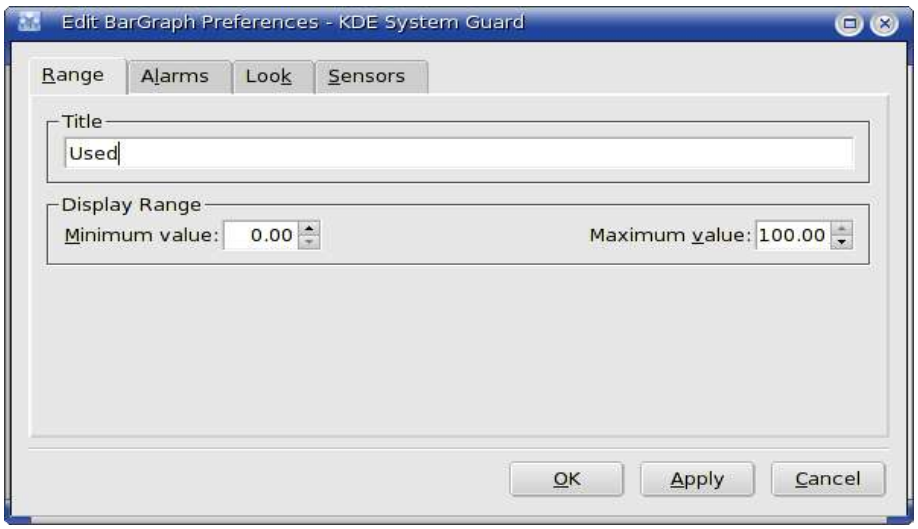


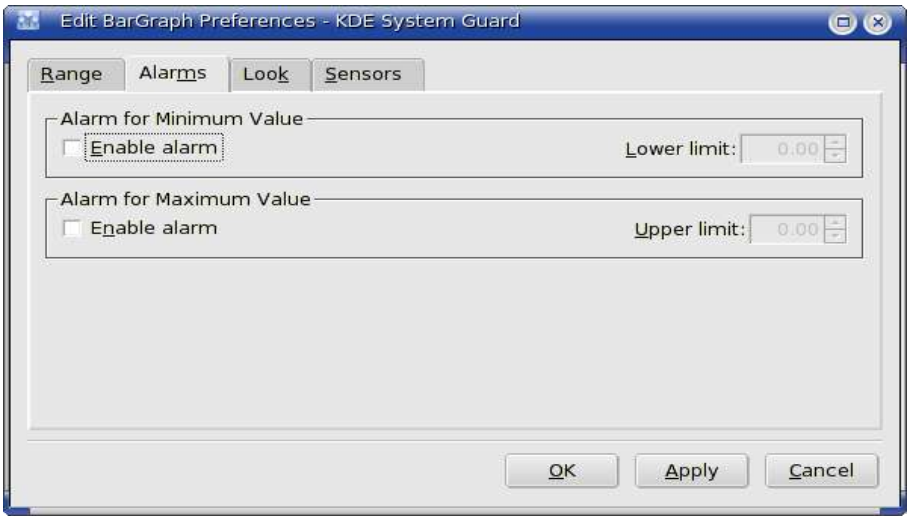
Notice that the unit is not noted here. Units for space are generally noted in KB's.

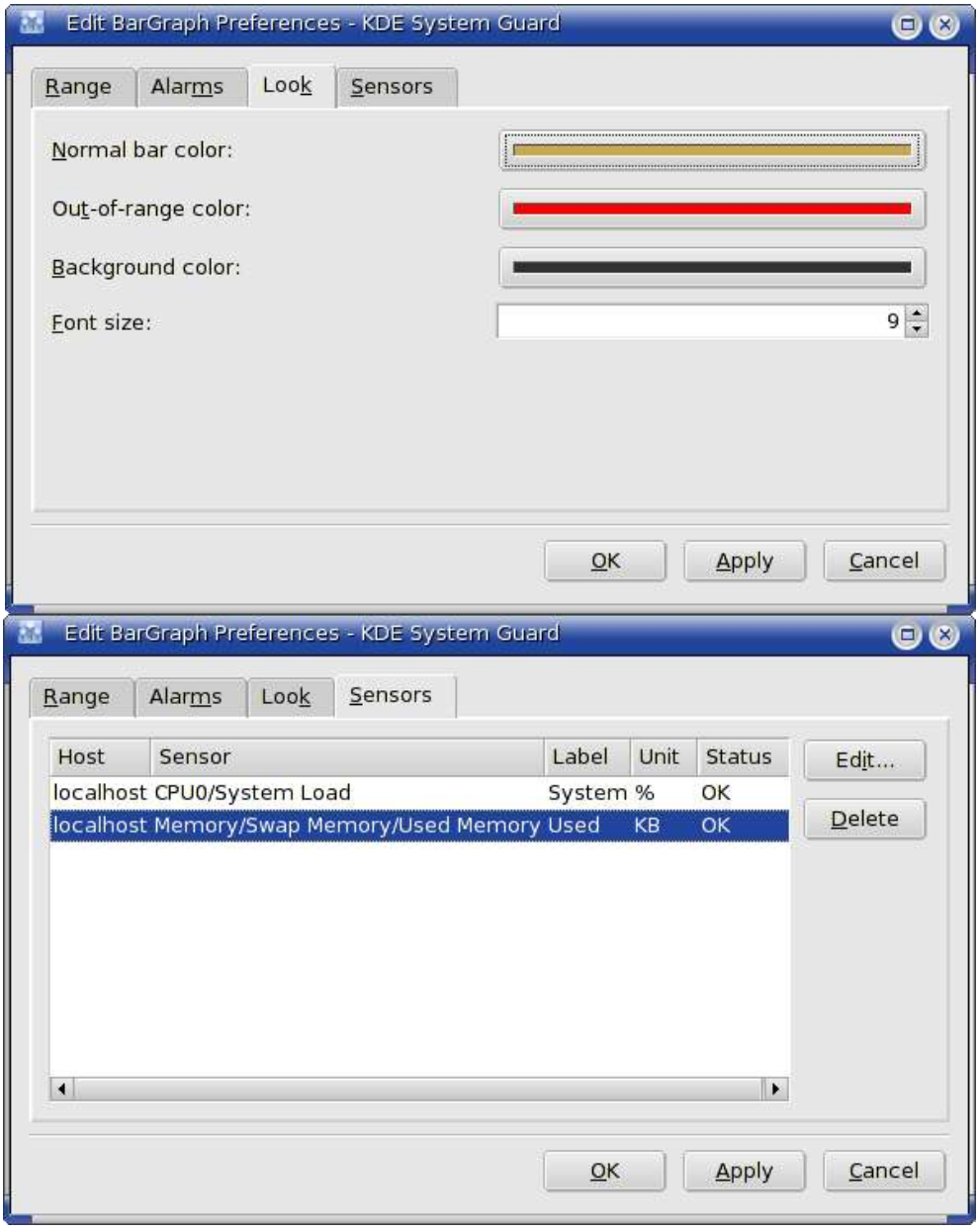
Here is the same information noted as a bar graph. In order to show you some differences, I also added CPU0 > System Load in the same display. Once a display type is selected, any additional sensors added will take on the same type.



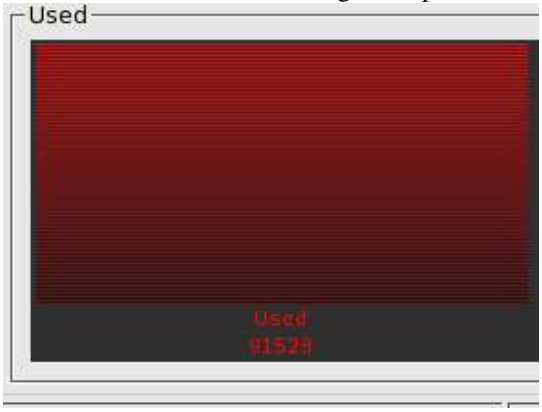
Notice how difficult it is to see the graph color against the dark background. This gives us a chance to look over the display properties and change this. Right-click in the display and select Properties. For this particular display, here are the four tabs (Range, Alarms, Look, and Sensors):



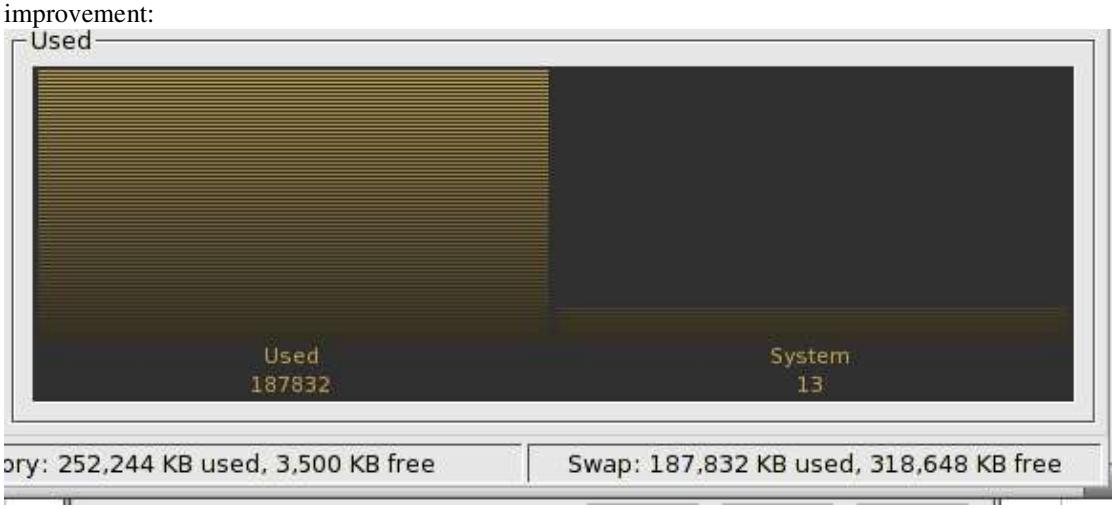




In Range, you can change the name of the display, and also what range to actually show.
In Alarms, you can set an alarm for a certain amount. There is a shortcoming to this, however. Notice that the Swap Used Space is up around 2,624KB. The alarm can only go between 0 and 100, showing it'd be best used for sensors showing percentages. Here is the display for Swap Space Used with the max at 100, definitely not what we want. This is something to keep in mind when setting an alarm.

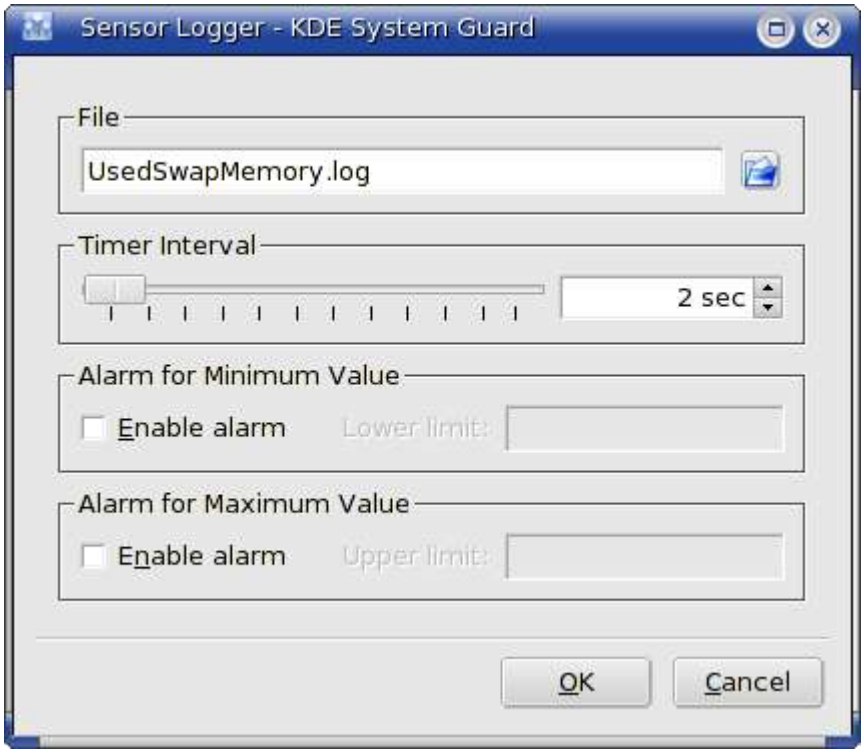


In Look, notice I've already changed the color for “Normal Bar Color”, and here is the remarkable improvement:

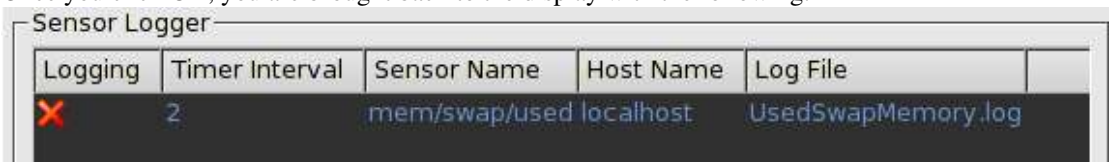


In Sensors, you are able to see what sensors populate this display, edit the label of the sensor, and delete a selected sensor.

The final type of display is SensorLogger, and this is probably the best tool for recording a baseline for a system (use Google to find different articles on recording a baseline for your servers and why this is important). Once you select SensorLogger, you are given a dialog box to enter a filename for the log, and to enter any alarms you'd want noted.



Once you click OK, you are brought back to the display with the following:



Notice we're not logging anything yet; to start logging, right-click anywhere on the line of the entry and select “Start Logging”. The X will turn into a green check mark. Here is the file after about 45 seconds of logging:

```
[kingttx@tomslaptop ~]$ cat UsedSwapMemory.log
Jan 10 12:42:40 localhost mem/swap/used: 88668
Jan 10 12:42:42 localhost mem/swap/used: 88668
Jan 10 12:42:44 localhost mem/swap/used: 88668
Jan 10 12:42:46 localhost mem/swap/used: 88668
Jan 10 12:42:48 localhost mem/swap/used: 88668
Jan 10 12:42:50 localhost mem/swap/used: 88668
Jan 10 12:42:52 localhost mem/swap/used: 88668
Jan 10 12:42:54 localhost mem/swap/used: 88668
Jan 10 12:42:56 localhost mem/swap/used: 88668
Jan 10 12:42:58 localhost mem/swap/used: 88668
Jan 10 12:43:00 localhost mem/swap/used: 88668
Jan 10 12:43:02 localhost mem/swap/used: 88668
Jan 10 12:43:04 localhost mem/swap/used: 88668
Jan 10 12:43:06 localhost mem/swap/used: 88668
Jan 10 12:43:08 localhost mem/swap/used: 88668
Jan 10 12:43:10 localhost mem/swap/used: 88668
Jan 10 12:43:12 localhost mem/swap/used: 88668
Jan 10 12:43:14 localhost mem/swap/used: 88668
Jan 10 12:43:16 localhost mem/swap/used: 88668
Jan 10 12:43:18 localhost mem/swap/used: 88668
Jan 10 12:43:20 localhost mem/swap/used: 88668
Jan 10 12:43:22 localhost mem/swap/used: 88668
```

Now we want to use this information and monitor another computer, such as a server or firewall. First, we connect to another host, preferably using SSH since it is the most secure. Click on File > Connect Host, or click the Connect Host button  .

In order for the remote connection to work, at least ksysguardd must be installed on the remote computer so that the local computer ksysguard client can pull the information from the remote computer. If your remote computer has KDE, most likely it will have ksysguardd (two d-s, the second for the daemon).

You'll be given the following dialog box:

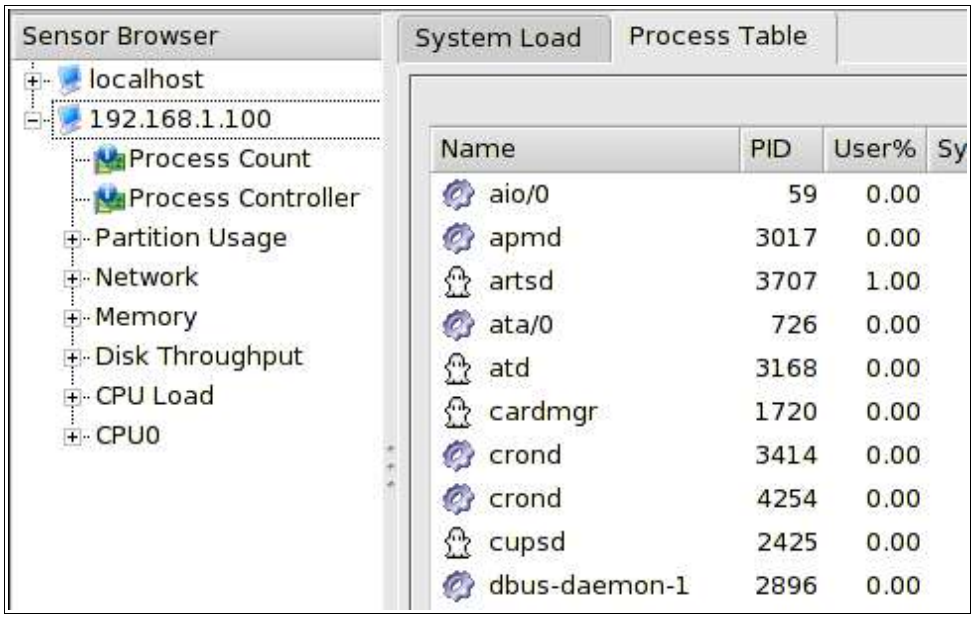
Put the host name or IP address of the system you want to monitor. Unless the host name can be resolved through /etc/hosts or some other means, you'll want to put the IP address of the remote computer instead. I do this by default since I have a small network anyway.

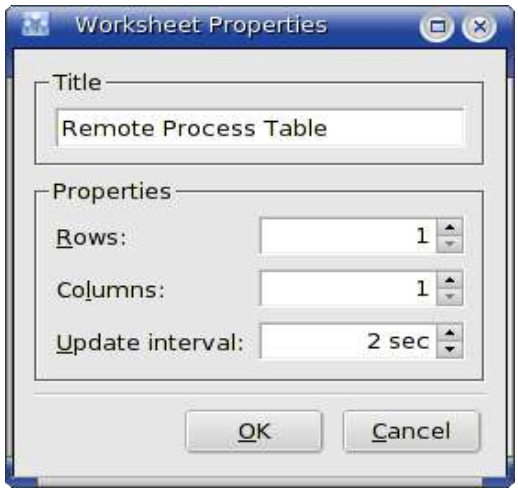
You'll notice the different connection types. Probably the most secure will be ssh, or a custom command calling an ssh session (see the example command in the screenshot above). The only issue is most ssh configurations disable root login by default to keep the system more secure, so the above command will not work in many cases. If you are using rsh, don't! It has too many security issues.

In this instance, I will use the “ssh” option. Once we've clicked OK, we are prompted for the password for SSH (unless you have listed the local host as trusted on the remote computer):

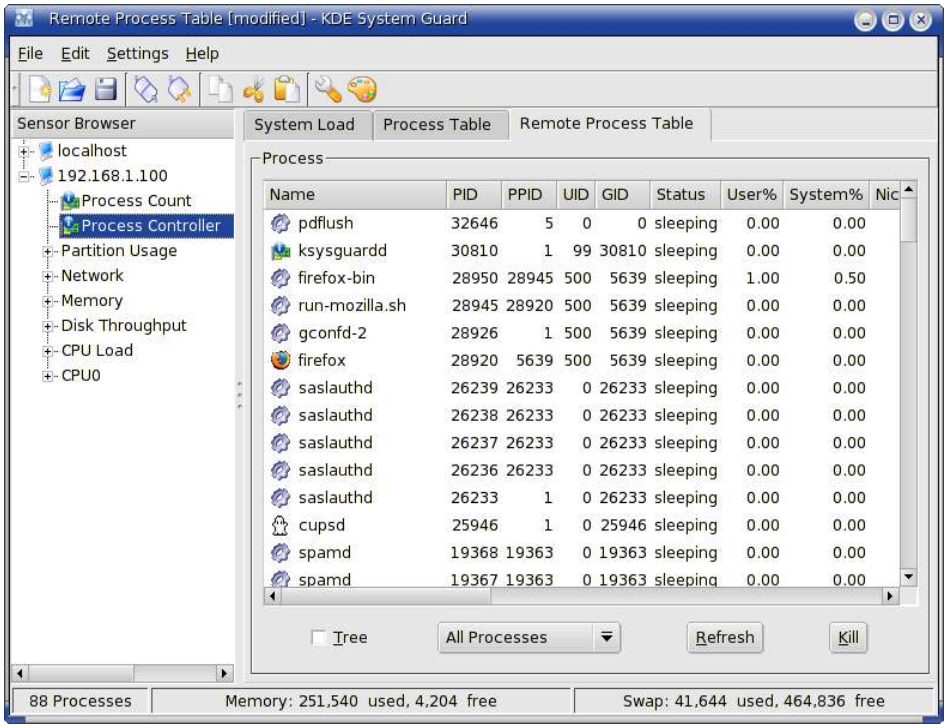
Once it connects, it'll take a little longer to bring in the available sensors for the remote host than what you'd expect for the localhost.

Here is what we have next, but you'll notice we don't have anything monitored from the remote host. Remember how we added a worksheet? Let's do that now by clicking the new worksheet icon. In this case, let's have a process table; we'll 1 row, 1 column, and let's label it something meaningful like “Remote Process Table”.





Now we have a blank worksheet, so let's make sure we are under the remote host and pull in Process Controller.



Again, remember the more sensors you have, and the quicker you have the updates, the more processor load you'll create, *and* since this is over the network, the more network traffic you'll create. Weigh your needs against this fact.

Once you're done, disconnect from the remote host by highlighting the host entry and click either the host disconnect button  or File > Disconnect Host. You'll notice any sensors for the remote host still show up, but the updates are frozen. If you need to keep the information visible, great! Otherwise, you can close that worksheet and either save or discard the worksheet when prompted.

Now that we've gone over the application and how to connect to a remote host, I'd like to stress the importance of keeping your eye on servers and creating a baseline so you won't be caught off-guard. True, I left that as some homework for you, the reader, to look up the importance of knowing the baseline for the systems you are responsible for. Once you know how the server behaves under a normal load and a normal amount of services running, you'll get a good idea what to look for in case your server begins to become bogged down, or the network is inundated (Denial of Service, anyone??), and you will be ready to react. Think of the ability to sit at your desk, run the sensors for disk space on your file server, and be able to report what is left to the powers that be.

I hope this helps and gives you a tremendous tool to use in Linux administration. Granted, there are some shortcomings with KsysGuard, and at the same time this is a powerful tool as it is.