

where the `sys/sysinfo.h` is available on LINUX, but not OSX or MS Windows. Before the system information can be retrieved, a struct variable of `sysinfo` type is created. This is not a simple variable, but contains several variables. The member variables of the struct are accessed using the `.` operator. When `sysinfo` is called, the address of the struct variable of `sysinfo` type is passed to the function. The function then writes the status into the member variables of the struct.

In the final example for this tutorial, `gnuplot` is used to plot the memory usage as a function of time:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/sysinfo.h>
int main() {
    int i, ramUsed;
    char gnuplotCmd[250], systemCmd[350];
    FILE *outPtr = 0;
    char fileName[50];
    sprintf(fileName,"data.txt"); /* The name of the output file. */
    struct sysinfo info; /* A sysinfo struct to hold the status. */
    outPtr = fopen(fileName,"w"); /* Open the output file. */
    if(!outPtr) return 1; /* If the output file cannot be opened return error */
    for(i=0;i<60;i++) {
        sysinfo(&info); /* Get the system information */
        ramUsed = info.totalram - info.freeram;
        fprintf(outPtr,"%d %d\n", i, ramUsed); /* Write the ram used. */
        usleep(500000); /* Sleep for 1/2 a second. */
    }
    fclose(outPtr); /* Close the output file. */

    /* Now plot the data */
    sprintf(gnuplotCmd, "plot '%s'\n", fileName); /* Build the plot command. */

    /* Create the full command, including the pipe to gnuplot */
    sprintf(systemCmd,"echo '%s' | gnuplot --persist",gnuplotCmd);

    system(systemCmd); /* Execute the system command. */
    return 0; /* Return success to the system. */
}
```

where the `sys/sysinfo.h` header file is available on LINUX and the `unistd.h` header file is available on LINUX or OSX. The program writes the memory usage to an output file every half a second. Then `gnuplot` is run to plot the memory usage as a function of time.

Challenge problem

Modify the previous example program to write an output file using the return value of the command `hostname` to form a file name. Then plot the memory used and the system load while running one or more other programs. The `loads[3]` member variable of the `sysinfo` struct holds the, one, five and fifteen minute load averages. Try using,

```
fprintf(outPtr,"%d %f %d\n", i, ramUsed/10240.0, info.loads[0]);
```

to write the data file. Then plot the data using two strings,

```
sprintf(gnuplotCmdOne, "plot '%s\' using 1:2 title '%s'", fileName, "Ram used");
sprintf(gnuplotCmdTwo, ", '%s\' using 1:3 title '%s'\n", fileName, "Load");

/* Create the full command, including the pipe to gnuplot */
sprintf(systemCmd,"echo '%s%s' | gnuplot -persist",gnuplotCmdOne,gnuplotCmdTwo);
```

The solution to the problem will be given next time.

Article by W. H. Bell