CSCI 352 Unix Software Development
Spring 2016
Assignment 3

Submitting Your Work

This assignment is worth 15% of the grade for the course. Save your program files (including header files and makefile) in a zipped tar file and submit the file via the **Assignment 3 Submission** item on the course web site. You must submit your assignment by 9:00am on Monday, May 30, 2016.

For this assignment, you are to implement a process monitor described here, providing a graphical user interface through GTK+3.0.

Your assignments will be evaluated on correct functionality and conformance to the coding standards described at the end of this assignment specification.

Process Monitor

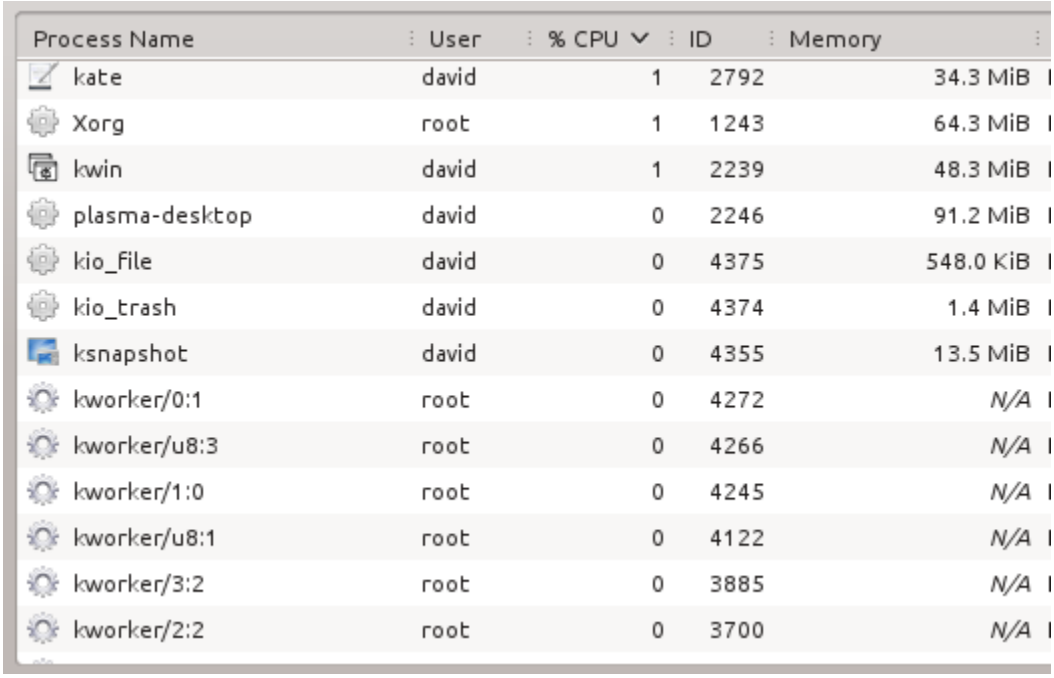
The performance monitor is to display process data in a scrollable GTK+ window.

Processes: A list of processes currently running on the system, showing for each process:

1. Name of the executable file, with an icon to represent the process.
2. The name of the user who ran the process.
3. The percentage of CPU time that the process is using.
4. The process ID
5. The amount of memory that the process is using.

The process list is to be sortable on any of these 5 columns.

The list must be refreshed every 1 second.



Process Name	User	% CPU	ID	Memory
kate	david	1	2792	34.3 MiB
Xorg	root	1	1243	64.3 MiB
kwin	david	1	2239	48.3 MiB
plasma-desktop	david	0	2246	91.2 MiB
kio_file	david	0	4375	548.0 KiB
kio_trash	david	0	4374	1.4 MiB
ksnapshot	david	0	4355	13.5 MiB
kworker/0:1	root	0	4272	N/A
kworker/u8:3	root	0	4266	N/A
kworker/1:0	root	0	4245	N/A
kworker/u8:1	root	0	4122	N/A
kworker/3:2	root	0	3885	N/A
kworker/2:2	root	0	3700	N/A

Coding Standards

1. Use meaningful names that give the reader a clue as to the purpose of the thing being named.
2. Avoid the repeated use of numeric constants. For any numeric constants used in your program, use #define preprocessor directives to define a macro name and then use that name wherever the value is needed.
3. Use comments at the start of the program to identify the purpose of the program, the author and the date written.
4. Use comments at the start of each function to describe the purpose of the function, the purpose of each parameter to the function, and the return value from the function.
5. Use comments at the start of each section of the program to explain what that part of the program does.
6. Use consistent indentation.