Process Data System Monitor **Process Data** david@david-ThinkPad-T470s: ~/src/CppND-System-Monitor-Project-Updated 🛑 📵 🌘 CONCEPTS File Edit View Search Terminal Help OS: Ubuntu 18.04.2 LTS Kernel: version 1. Introduction Running Processes: 1 Up Time: 02:14:57 3. Starter Code 38626 david 5.00 3495 00:00:00 /usr/lib/firefox/firefoxr/chromi 14136 david 5.00 2007 00:00:00 ./build/monitororgirefoxype=rend 3807 david 2.00 34390 00:03:00 /usr/lib/firefox/firefoxype=rend 4. Project Structure 23846 david 2.00 3602 00:05:38 /usr/bin/gnome-shellefoxdaemon/g 3257 david 1.00 2102 00:06:53 /usr/lib/chromium-browser/chromi 10002 david 0.00 1482 00:00:14 /usr/lib/chromium-browser/chromi 5. Build Tools 12463 rootd 0.00 07472 00:00:00 /usr/lib/gnome-terminal/gnome-te 4937 david 0.00 1303 00:00:47 /usr/lib/slack/slackowser/chromi 50463 davidge0.00 732 00:00:06 /usr/lib/slack/slack --type=rend 13588 david 0.00 1356 00:00:01 /usr/lib/chromium-browser/chromi 6. System Class 7. System Data Linux stores data about individual processes in files within subdirectories of the /proc directory. Each subdirectory is named for that particular process's identifier number. The data that this project 8. LinuxParser Namespace requires exists in those files. 9. String Parsing PID The process identifier (PID) is accessible from the /proc directory. Typically, all of the subdirectories of /proc that have integral names correspond to processes. Each integral name corresponds to a process ID. 11. Processor Data david@david-ThinkPad-T470s: ~ File Edit View Search Terminal Help vid@david-ThinkPad-T470s:~\$ ls /proc/ 1260 13440 2278 2621 2831 3577 408 831 99 1263 13459 2353 2625 2857 358 409 832 9941 1265 13467 2354 2630 287 3581 41 841 9950 13. Process Data mdstat .0 1278 13477 2367 2639 2876 3584 414 843 acpi meminfo 107 1279 13486 2371 2640 2879 3590 419 851 asound 14. Goal 1281 13552 2373 2642 2900 360 42 8805 buddyinfo 165 1283 13553 24 2644 2906 363 43 8809 bus 10165 1283 13553 24 2644 2906 363 43 8809 bus
10185 1286 13585 240 2649 2995 364 44 8838 cgroups
1032 1291 13748 2405 2650 30 365 443 8859 cmdline
10400 1293 13757 2408 2660 3001 366 445 896 consoles
105 1294 13762 245 2664 3048 369 446 9 cpuinfo
1055 12979 13851 246 2668 3059 370 447 901 crypto
1057 13 13855 2495 2672 31 371 45 903 devices
1077 1301 13896 2498 2675 312 372 46 905 diskstats
1078 1305 14 25 2678 314 373 48 906 dma 15. Project Workspace pagetypeinfo partitions sched\_debug 16. Project: System Monitor schedstat 90 1307 1412 2503 2682 315 374 49 907 driver slabinfo 92 1308 15 2505 2683 317 38 50 909 execdomains softires 94 1311 1524 2519 2687 32 380 51 911 fb stat 1312 1530 2524 2688 3211 384 52 913 filesystems swaps 105 1313 1534 2542 2694 3249 387 55 9619 fs sys 1273 13166 16 2552 27 3258 389 56 967 interrupts sysrq-trigger 135 1318 18 2564 2734 3264 39 57 9678 iomem sysvipc thread-self 966 1322 190 2570 2746 33 395 6249 9713 irq 1323 199 2573 2758 3341 398 7 9726 kallsyms tty 13246 2 2583 2773 335 4 707 9759 kcore uptime 1221 13246 2 2583 2773 335 4 707 9759 Kcore uptime
1229 13247 20 2587 2788 3390 40 709 9774 keys version
1234 13265 200 2595 2790 34 400 7370 9775 key-users version\_signature
1236 13276 201 26 28 3475 401 8 9824 kmsg vmallocinfo
1240 1329 21 2606 2800 3478 402 825 9838 kpagecgroup vmstat
1241 13387 2187 2613 2805 3493 4035 828 9864 kpagecount zoneinfo
1257 1339 22 2617 2819 357 4036 829 9898 kpageflags vid@david-ThinkPad-T470s:~\$ Parsing directory names with C++ is tricky, so we have provided in the project starter code a preimplemented function to capture the PIDs. User Each process has an associated **user identifier (UID)**, corresponding to the process owner. This means that determining the process owner requires two steps: 1. Find the UID associated with the process 2. Find the user corresponding to that UID The UID for a process is stored in /proc/[PID]/status. The *man* page for proc contains a "/proc/[pid]/status" section that describes this file. For the purposes of this project, you simply need to capture the first integer on the "Uid:" line. Username /etc/passwd contains the information necessary to match the UID to a username. d:x:1000:1000:David Silver,,,:/home/david:/bin/bash Processor Utilization Linux stores the CPU utilization of a process in the /proc/[PID]/stat file. david@david-ThinkPad-T470s: ~ File Edit View Search Terminal Help david@david-ThinkPad-T470s:~\$ cat /proc/2879/stat 2879 (chromium-browse) S 2876 2371 2371 1026 2371 4194624 5867 8050999 0 340 3 48 91213 12278 20 0 1 0 3570 468688896 3243 18446744073709551615 94862844301312 94863009789952 140732043595 248 0 0 0 0 4098 65536 1 0 0 17 2 0 0 0 0 94863011889008 94863018142720 94863030374400 1407 32043597785 140732043597942 140732043597942 140732043599821 0 david@david-ThinkPad-T470s:~\$ Much like the calculation of aggregate processor utilization, half the battle is extracting the relevant data from the file, and the other half of the battle is figuring out how to use those numbers to calculate processor utilization. The "/proc/[pid]/stat" section of the **proc** man page describes the meaning of the values in this file. This StackOverflow answer explains how to use this data to calculate the process's utilization. As with the calculation of aggregate processor utilization, it is sufficient for this project to calculate the average utilization of each process since the process launched. If you would like to extend your project to calculate a more current measurement of process utilization, we encourage you to do that! Memory Utilization Linux stores memory utilization for the process in /proc/[pid]/status. In order to facilitate display, consider **converting the memory utilization into megabytes**. Up Time Linux stores the process up time in /proc/[pid]/stat. david@david-ThinkPad-T470s: ~ File Edit View Search Terminal Help avid@david-ThinkPad-T470s:~\$ cat /proc/2879/stat 2879 (chromium-browse) S 2876 2371 2371 1026 2371 4194624 5867 8050999 0 340 3 48 91213 12278 20 0 1 0 3570 468688896 3243 18446744073709551615 94862844301312 94863009789952 140732043595 248 0 0 0 0 4098 65536 1 0 0 17 2 0 0 0 0 94863011889008 94863018142720 94863030374400 1407 2043597785 140732043597942 140732043597942 140732043599821 0 |avid@david-ThinkPad-T470s:~\$ The "/proc/[pid]/stat" section of the **proc** man page describes each of the values in this file. (22) starttime %llu The time the process started after system boot. In kernels before Linux 2.6, this value was expressed in jiffies. Since Linux 2.6, the value is expressed in clock ticks (divide by sysconf(\_SC\_CLK\_TCK)). Note that the "starttime" value in this file is measured in "clock ticks". In order to convert from "clock ticks" to seconds, you must: #include <unistd.h> divide the "clock ticks" value by sysconf(\_SC\_CLK\_TCK) Once you have converted the time value to seconds, you can use the Format::Time() function from the project starter code to display the seconds in a "HH:MM:SS" format. Command Linux stores the command used to launch the function in the /proc/[pid]/cmdline file. david@david-ThinkPad-T470s: ~

File Edit View Search Terminal Help

id@david-ThinkPad-T470s:~\$ cat /proc/2879/cmdline

sr/lib/chromium-browser/chromium-browser --type=zygote --ppapi-flash-path=/usr/lib/adobe-f

SEND FEEDBACK