Aidan Dowling | Trishita Bhattacharya

CMPSC 473

**LAB 2 REPORT**

**Task 3.2.1**

*Output for lmbench testbench:*

Context switching - times in microseconds - smaller is better

-------------------------------------------------------------------------

|  |  |  |
| --- | --- | --- |
| **Host** | **OS** | **2p/0K ctxsw (us)** |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.12 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 6.90 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.18 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.12 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.29 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 6.90 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.18 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.01 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.12 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.29 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 6.97 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 6.90 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.12 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.29 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 6.97 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 6.50 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 6.90 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.18 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.01 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 6.90 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 6.90 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.18 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.12 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 6.90 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.18 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.01 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.12 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.29 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 6.90 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.18 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.01 |
| cse-p204inst10.cse.psu.edu | Linux 2.6.32 | 7.83 |

Average = 7.31 us Variance = 0.0467

*Output for User Level Thread counter():*

|  |  |
| --- | --- |
| **No. context switches** | **Context Switch Overhead (us)** |
| 12 | 2.165 |
| 12 | 2.235 |
| 12 | 2.375 |
| 12 | 2.305 |
| 12 | 2.584 |
| 12 | 2.304 |
| 12 | 2.933 |
| 12 | 2.305 |
| 12 | 2.794 |
| 12 | 2.375 |
| 12 | 2.374 |
| 12 | 2.375 |
| 12 | 2.235 |
| 12 | 2.375 |
| 12 | 2.794 |
| 12 | 2.235 |
| 12 | 2.934 |
| 12 | 2.514 |
| 12 | 2.305 |
| 12 | 2.305 |
| 12 | 2.305 |
| 12 | 2.445 |
| 12 | 2.374 |
| 12 | 2.304 |
| 12 | 2.235 |
| 12 | 2.305 |
| 12 | 2.305 |
| 12 | 2.584 |
| 12 | 2.305 |
| 12 | 2.305 |
| 12 | 2.165 |
| 12 | 2.304 |

Average = 2.33 us Variance = 0.0418

*Comparing output for Kernel Level and User Level Thread Context Switch Overhead:*

**Task 3.2.2**

*Output for Kernel Level Thread sleeping():*

Code under execution ..

|  |  |  |
| --- | --- | --- |
| arr[0] = 278432  arr[1] = 42558  arr[2] = 238702  arr[3] = 43278  arr[4] = 237857  arr[5] = 42100  arr[6] = 238742  arr[7] = 42185  arr[8] = 240013  arr[9] = 42525  arr[10] = 237995  arr[11] = 45947  arr[12] = 237965  arr[13] = 43448  arr[14] = 234857  arr[15] = 42730  arr[16] = 234904  arr[17] = 42271  arr[18] = 234551  arr[19] = 45206  arr[20] = 235022  arr[21] = 46023  arr[22] = 234760  arr[23] = 46796  arr[24] = 234482  arr[25] = 0  arr[26] = 0  arr[27] = 0  arr[28] = 0  arr[29] = 0  arr[30] = 0  arr[31] = 0  arr[32] = 0  arr[33] = 0 | arr[34] = 0  arr[35] = 0  arr[36] = 0  arr[37] = 0  arr[38] = 0  arr[39] = 0  arr[40] = 225334  arr[41] = 49857  arr[42] = 307606  arr[43] = 40621  arr[44] = 228945  arr[45] = 39914  arr[46] = 227332  arr[47] = 44406  arr[48] = 252761  arr[49] = 43245  arr[50] = 252604  arr[51] = 43716  arr[52] = 226359  arr[53] = 43062  arr[54] = 224690  arr[55] = 40715  arr[56] = 228318  arr[57] = 41323  arr[58] = 251413  arr[59] = 41596  arr[60] = 253309  arr[61] = 41548  arr[62] = 261429  arr[63] = 39869  arr[64] = 264836  arr[65] = 42653  arr[66] = 281070  arr[67] = 41387 | arr[68] = 278049  arr[69] = 40627  arr[70] = 279728  arr[71] = 41244  arr[72] = 280541  arr[73] = 40426  arr[74] = 264042  arr[75] = 43151  arr[76] = 265881  arr[77] = 38602  arr[78] = 248016  arr[79] = 38872  arr[80] = 262945  arr[81] = 39761  arr[82] = 263794  arr[83] = 39588  arr[84] = 265223  arr[85] = 38514  arr[86] = 278638  arr[87] = 38782  arr[88] = 262403  arr[89] = 46763  arr[90] = 262444  arr[91] = 38802  arr[92] = 278728  arr[93] = 46273  arr[94] = 279588  arr[95] = 45306  arr[96] = 290289  arr[97] = 44600  arr[98] = 280098  arr[99] = 45279 |

Counter = 12730264

*Output for User Level Thread sleeping():*

|  |  |
| --- | --- |
| **Thread No.** | **Counter value** |
| 1 | 9498 |
| 2 | 30590 |
| 3 | 22000 |
| 4 | 10686 |
| 5 | 11035 |
| 6 | 25841 |
| 7 | 11594 |
| 8 | 29892 |
| 9 | 12223 |
| 10 | 30730 |
| 11 | 30800 |
| 12 | 20533 |
| 13 | 33175 |
| 14 | 22558 |
| 15 | 33873 |
| 16 | 22978 |
| 17 | 34012 |
| 18 | 41695 |
| 19 | 34781 |
| 20 | 22558 |
| 21 | 19415 |
| 22 | 42743 |
| 23 | 14178 |
| 24 | 43790 |
| 25 | 14108 |
| 26 | 48750 |
| 27 | 14457 |
| 28 | 22279 |
| 29 | 37924 |
| 30 | 15365 |
| 31 | 14806 |
| 32 | 35758 |
| 33 | 14178 |
| 34 | 18927 |
| 35 | 43650 |
| 36 | 15365 |
| 37 | 37295 |
| 38 | 21721 |
| 39 | 15435 |
| 40 | 15296 |
| 41 | 15295 |
| 42 | 42114 |
| 43 | 13968 |
| 44 | 13968 |
| 45 | 35758 |
| 46 | 19066 |
| 47 | 36178 |
| 48 | 13689 |
| 49 | 35130 |
| 50 | 17950 |
| 51 | 13269 |
| 52 | 38343 |
| 53 | 13340 |
| 54 | 35619 |
| 55 | 12991 |
| 56 | 12921 |
| 57 | 32546 |
| 58 | 34571 |
| 59 | 12013 |
| 60 | 31080 |
| 61 | 11943 |
| 62 | 30939 |
| 63 | 11593 |
| 64 | 31778 |
| 65 | 32337 |
| 66 | 20534 |
| 67 | 10826 |
| 68 | 28286 |
| 69 | 9428 |
| 70 | 23257 |
| 71 | 9079 |
| 72 | 25702 |
| 73 | 9638 |
| 74 | 9149 |
| 75 | 20254 |
| 76 | 10965 |
| 77 | 18648 |
| 78 | 10546 |
| 79 | 16972 |
| 80 | 9219 |
| 81 | 16063 |
| 82 | 12502 |
| 83 | 5866 |
| 84 | 6845 |
| 85 | 7039 |

*Comparing output for Kernel Level and User Level Thread - sleeping():*

Kernel Level threads seem to handle the sleeping() function much better than the User Level threads. This is most likely because the calls to sleep() within the threads created a blocking call situation. The sleep calls stop the program and doesn’t hand it off to other threads.

**Task 3.2.3**

*Output for varying no. of threads when based on Kernel Level Thread counter():*

|  |  |
| --- | --- |
| **No. of Threads** | **Total Counter Value** |
| 1 | 17684012 |
| 2 | 17440055 |
| 4 | 17947912 |
| 8 | 16157344 |
| 16 | 17788046 |
| 32 | 17188676 |

*Output for varying no. of threads when based on User Level Thread counter():*

|  |  |
| --- | --- |
| **No. of Threads** | **Total Counter Value** |
| 1 | 98218609 |
| 2 | 106701096 |
| 4 | 123409847 |
| 8 | 140155125 |
| 16 | 226370925 |
| 32 | 354235656 |

*Comparing output for varying no. of threads when based on Kernel Level and User Level Thread – counter():*