Project 2 Report: POSIX Thread Programming
CSE 3320:Operating Systems
GROUP 3

Hamilton Nguyen 1000538439

March 25, 2020

Date Performed:	March 15, 2020
Partners:	Marvin Willington 1001660133

1 Assignment 1: Substring

According to the specifications stated in assignment 1 for project 2, all steps were carried out successfully on a 64-bit personal computer using a ubuntu via virtual box. The NUMTHREADS in the source code is setted at 4. As per request of the specification, the local number is added into a global variable which shows the total number of matched substrings in string s1. Refer to figure 1.

2 Assignment 2: Condition variables

According to the specifications stated in assignment 2 for project 2, all steps were carried out successfully on a 64-bit personal computer using a ubuntu via virtual box. The implementation of the producer-consumer algorithm using condition variables in the source code can be found directly in the source file folder. As per specification, the buffer (queue) size of 5 characters is setted in the source code. Refer to figure 2.

3 Assignment 3: Quantification of context switch from two micro-benchmarks

According to the specifications stated in assignment 3 for project 2, all steps were carried out successfully on a 64-bit personal computer using a ubuntu via virtual box. Two micro benchmarks source codes, ProcessWOSwitch C file and ProcessWSwitch C file, were created to quantify the total costs of context switch between multiple processes and multiple threads. The lat ctx benchmark from

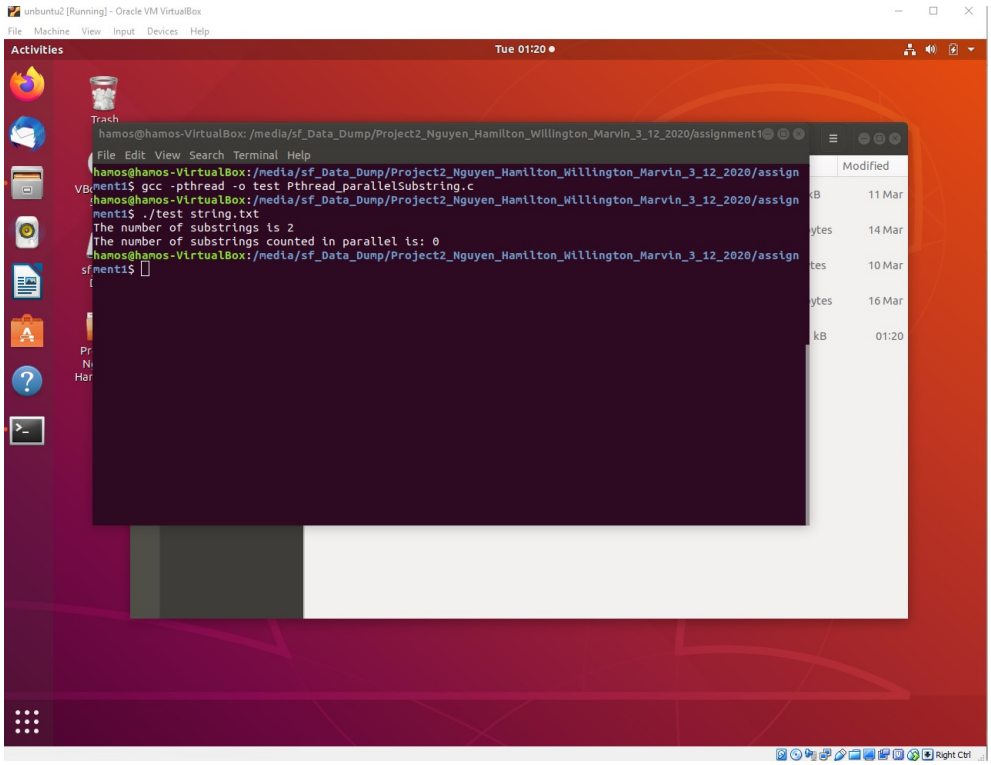

Figure 1: Assignment 1: print out number of finding the substring using PThread method

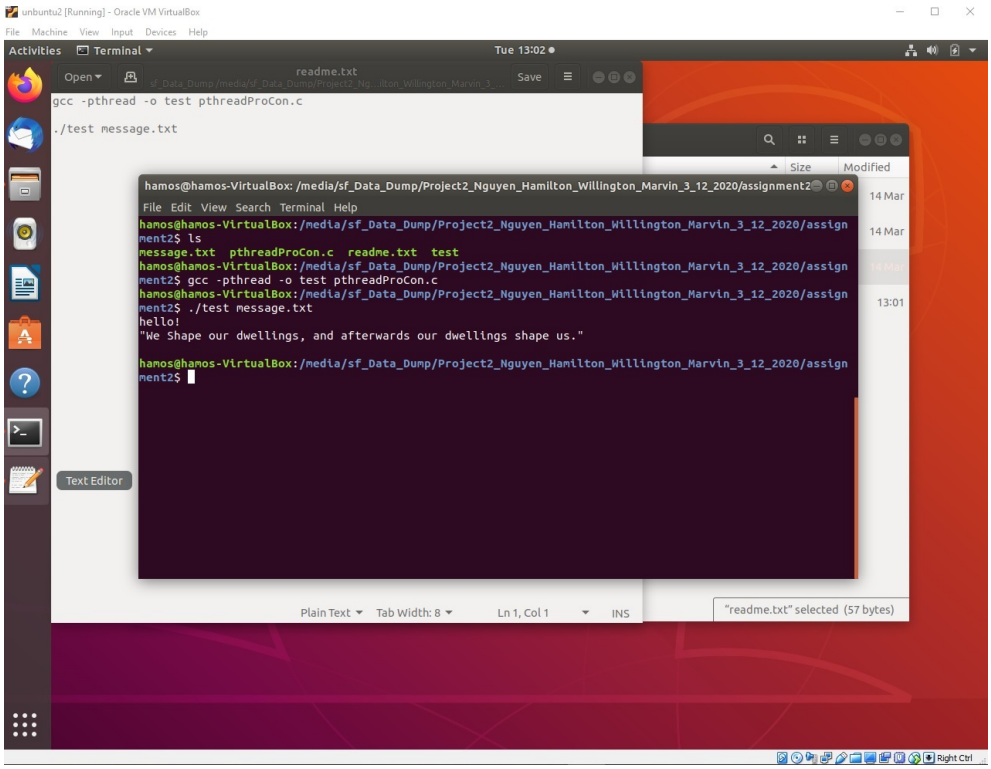

Figure 2: Assignment 2: Printout of Pthread program using condition variables

Dual Core Processor				
n	s	Measure Single Process (uSec)	Measure Switch (uSec)	Total Cost of Context Switch
0	0	1.578495009	3.76892278	2.190427771
0	64	1.683262543	3.819044314	2.135781771
10	64	1.552855035	4.156399566	2.603544531
1000	64	1.729013976	4.105016211	2.376002235
2048	64	2.00813342	4.112341515	2.104208095
10000	64	3.225642231	5.649961328	2.424319097
100000	64	24.7481645	28.57415809	3.825993598
256000	64	58.63078442	77.21490786	18.58412344
500000	64	182.2581738	197.4635894	15.20541558
1000000	64	393.7315755	471.091408	77.35983244
2048000	64	1013.269804	1310.723767	297.4539635

Figure 3: Assignment 3: Measured data of Context Switch of Dual Core Processor, n is ArraySize and S is the StrideSize.

Triple Core Processor				
n	s	Measure Single Process (uSec)	Measure Switch (uSec)	Total Cost of Context Switch
0	0	1.457002648	3.839564475	2.382561827
0	64	1.484910503	3.853311089	2.368400586
10	64	1.484255382	4.034228542	2.54997316
1000	64	1.637008507	4.025830252	2.388821745
2048	64	2.023952995	4.486942274	2.46298928
10000	64	3.049557118	6.020658594	2.971101476
100000	64	24.04156076	27.64232474	3.600763976
256000	64	70.82532556	82.42064392	11.59531836
500000	64	183.5640578	198.328484	14.76442624
1000000	64	402.6051536	467.2248544	64.6197008
2048000	64	986.6004855	1472.687057	486.0865714

Figure 4: Assignment 4: Measured data of Context Switch of triple Core Processor, n is ArraySize and S is the StrideSize.

lmbench benchmark were used as a reference about how to measure the context switch cost between multiple processes. The ProcessWOSwitch C file is a single process simulating two process communcations, read and write, without accounting context switch. While the ProcessWSwitch C file simulates a read and write communications through a pipe and accounts for context switching. The Functions C and Header file is a collection of functions that are used to measure context switch cost. In order to compile this collection of files, a makefile was created for this assignment. The virtualbox application were configured to the various number of levels vCPUs that is more than one and all processes/threads are run on a single level vCPU. Refer to figure 3, 4, 5, 6, 7, 8, and 9. Note to calculate total time in microseconds (usecs) determine time1 and time2 and let (total time context switch = time2-time1).

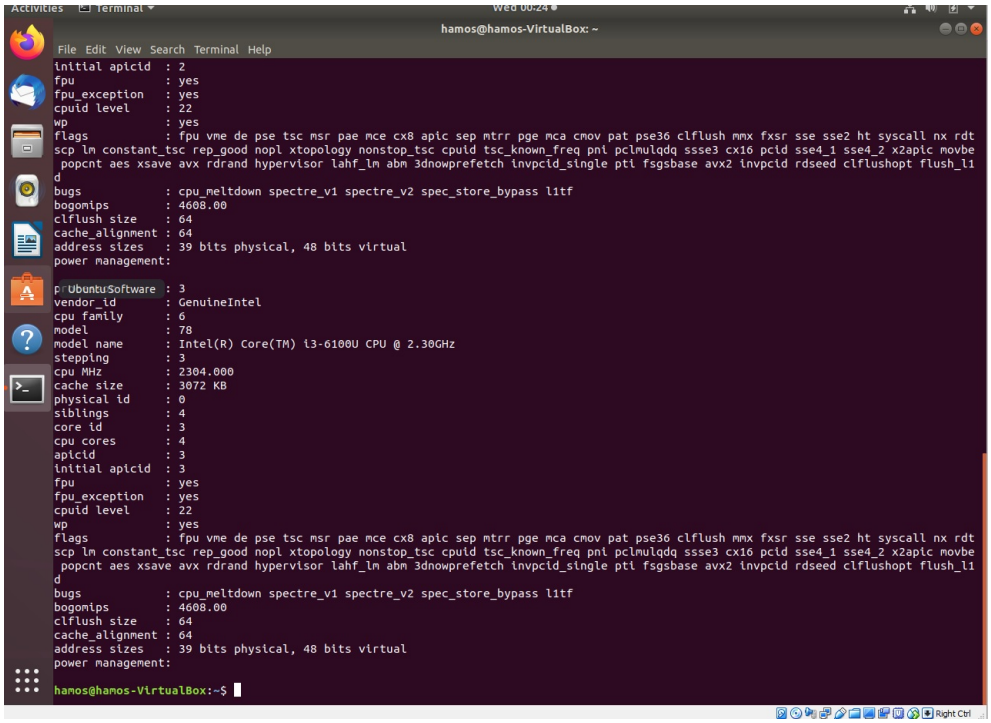

Figure 5: Assignment 5: Screenshot of setting VCPUs to simulate 4 cores.

Quad Core Processor				
n	s	Measure Single Process (uSec)	Measure Switch (uSec)	Total Cost of Context Switch
0	0	1.587891753	3.804108203	2.21621645
0	64	1.559864844	4.042978299	2.483113455
10	64	1.602119097	3.873515625	2.271396528
1000	64	2.189749566	4.033393707	1.843644141
2048	64	1.776568576	4.242584852	2.466016276
10000	64	3.11979783	5.387370486	2.267572656
100000	64	24.36913932	28.39556875	4.026429427
256000	64	74.96927248	76.71037018	1.7410977
500000	64	181.9190977	201.3775946	19.45849683
1000000	64	393.0728826	468.8861919	75.81330924
2048000	64	1077.243993	1287.729937	210.4859438

Figure 6: Assignment 6: Measured data of Context Switch of Quad Core Processor, n is ArraySize and S is the StrideSize.

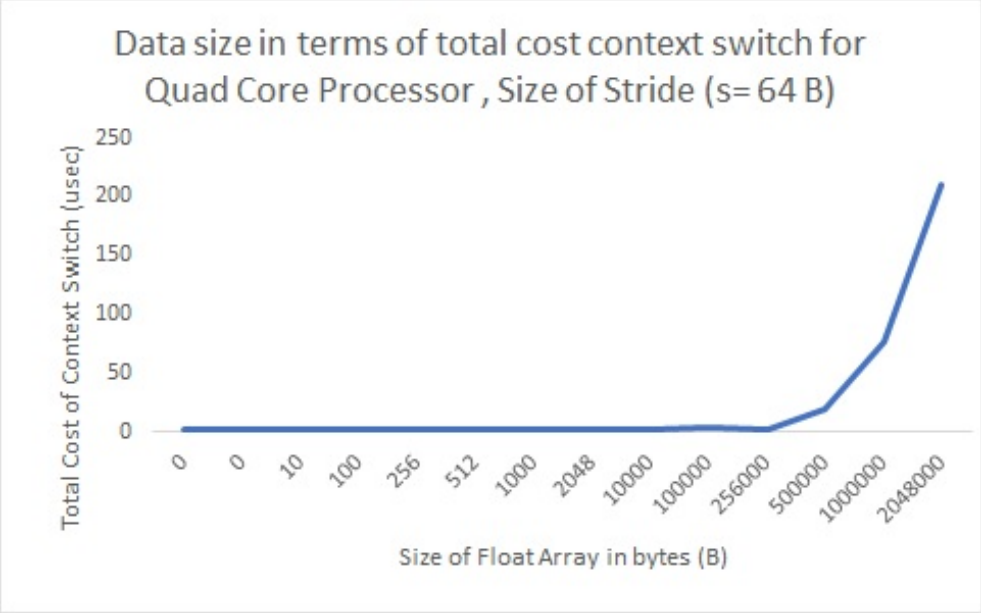

Figure 7: Assignment 7: Data plot of various Data size in terms of total cost context switch for Quad Core Processor.

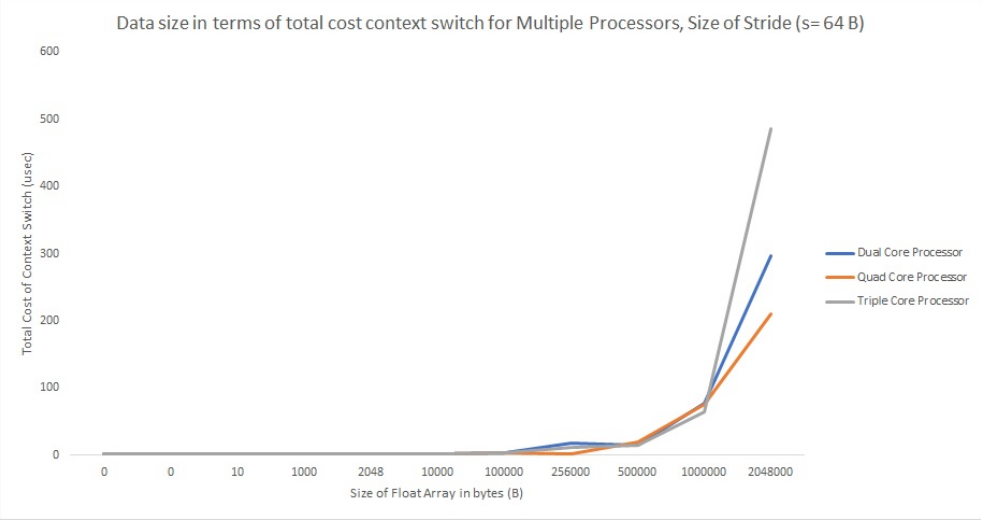

Figure 8: Assignment 8: Data size in terms of total cost context switch for various VCPUs settings.

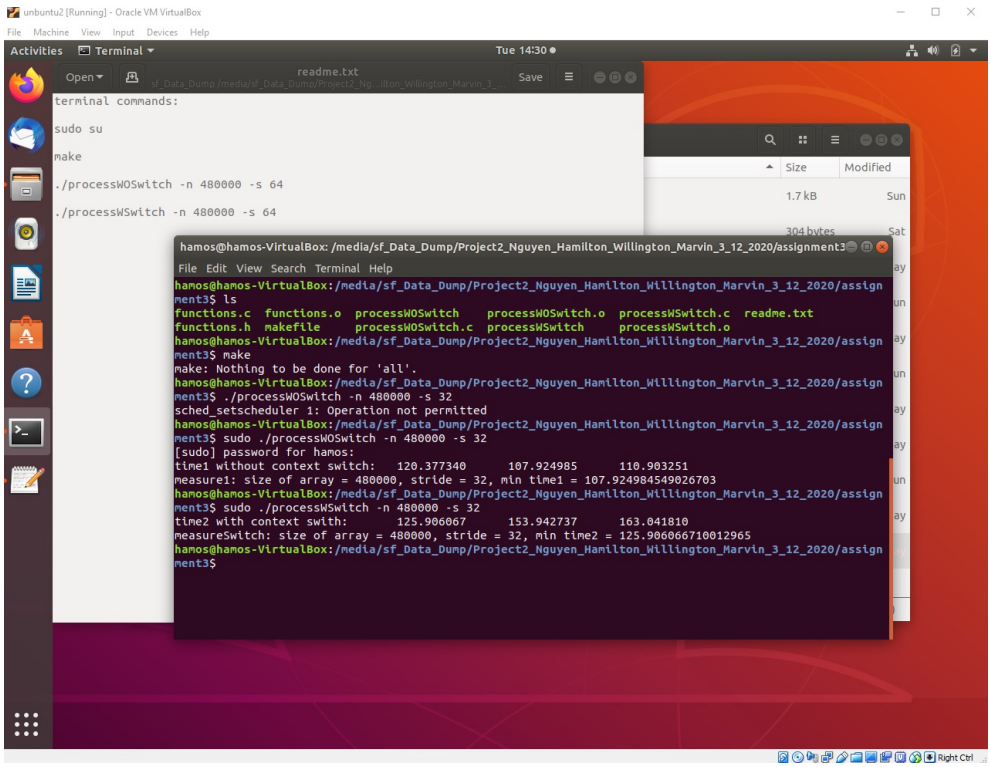

Figure 9: Assignment 9: Compiler execution of system files for quantification of context switch to determine time1 and time2.