

Bedirhan Bardakci - 20150702053

12.05.2021

TABLE OF CONTENTS

ABSTRACT	1
Process vs Threads	2
General Performance Comparison Between Threads vs Processes	2
DESING & IMPLEMENTATION	3
TEST AND RESULTS	4
10 file & 10 Kb Buffer	4
10 File & 105 Kb Buffer	5
10 File & 1000 Kb Buffer	6
CONCLUSION	9
REFERENCES	10

ABSTRACT

POSIX semaphores allow processes and threads to synchronize their actions. Actually, a semaphore is an integer whose value is never allowed to fall below zero. Two operations can be performed on semaphores: increment the semaphore value by one (sem_post); and decrement the semaphore value by one (sem_wait)

Shared Memory is an efficient way of passing data between processes. One process will create a memory portion which other processes (if permitted) can access.

Process vs Threads

Processes vs. Threads — Advantages and Disadvantages

PROCESS	THREAD
Processes are heavyweight operations	Threads are lighter weight operations
Each process has its own memory space	Threads use the memory of the process they belong to
Inter-process communication is slow as processes have different memory addresses	Inter-thread communication can be faster than inter-process communication because threads of the same process share memory with the process they belong to
Context switching between processes is more expensive	Context switching between threads of the same process is less expensive
Processes don't share memory with other processes	Threads share memory with other threads of the same process

General Performance Comparison Between Threads vs Processes

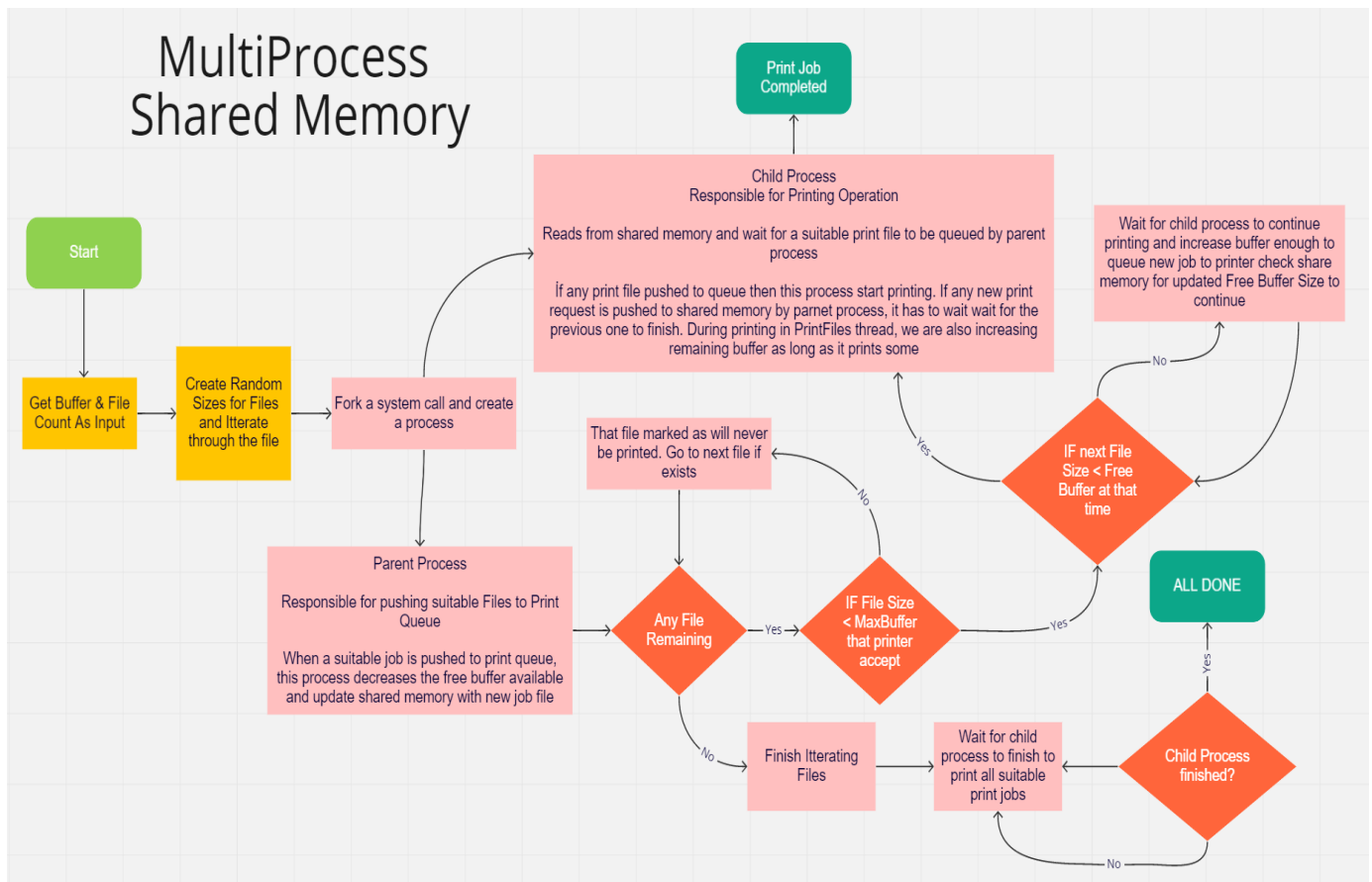
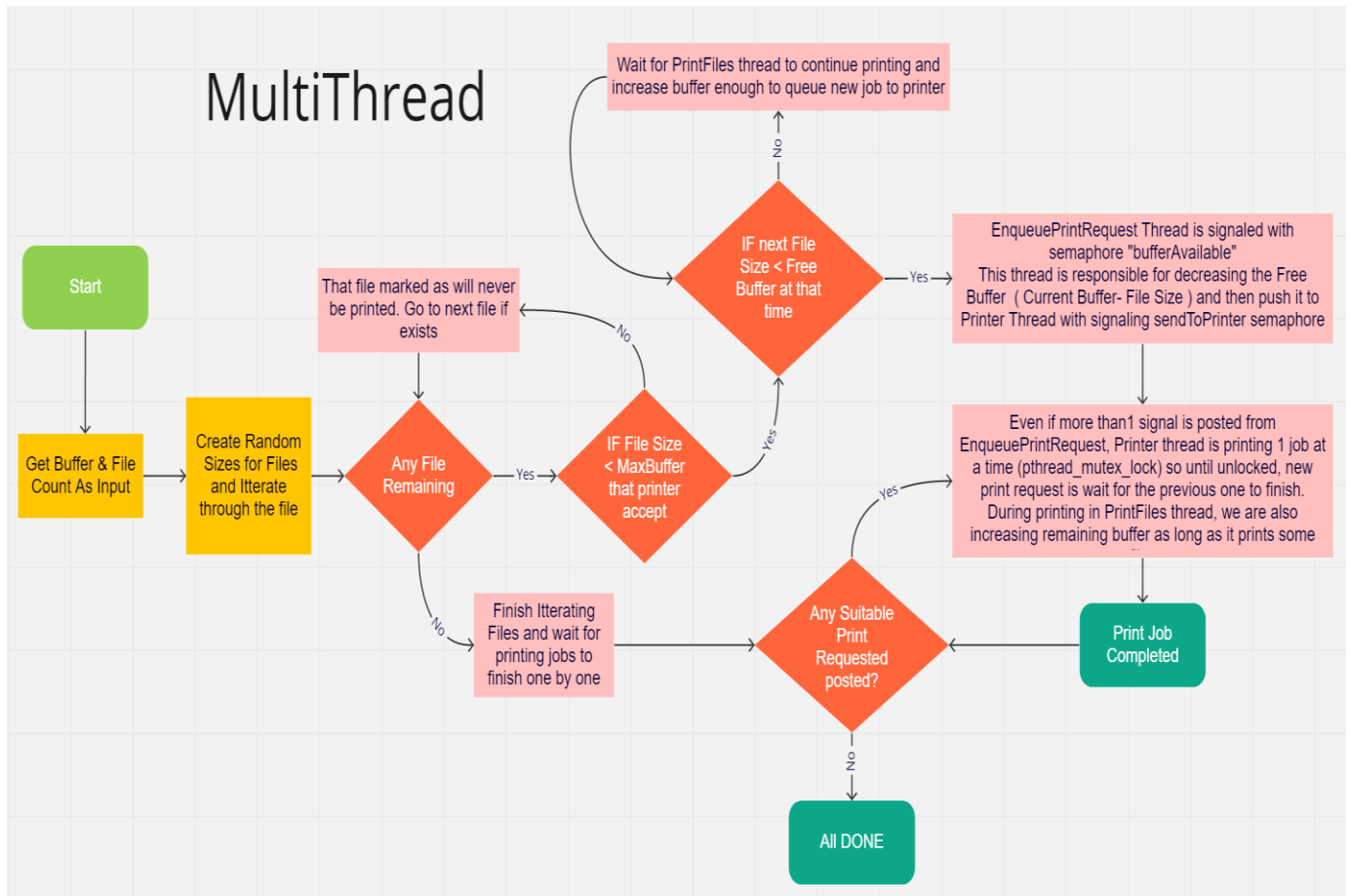
That depends on a lot of factors. Processes are more heavy-weight than threads, and have a higher startup and shutdown cost. Creating many processes will consume more memory. Each process takes up memory Inter-process communication (IPC) is also harder and slower than interthread communication.

Conversely, processes are safer and more secure than threads, because each process runs in its own virtual address space. If one process crashes or has a buffer overrun, it does not affect any other process at all, whereas if a thread crashes, it takes down all of the other threads in the process, and if a thread has a buffer overrun, it opens up a security hole in all of the threads.

During coding, working with process is take much more time of me, than working with threads because there is an implementation need for shared memories etc... As a result MultiProcess project has 100 lines more than MultiThread project of mine.

So I think working with threads is much more easier.

DESING & IMPLEMENTATION



TEST AND RESULTS

10 file & 10 Kb Buffer

No File is smaller than the Buffer so Total Print = 0

```
Printer Spooler ( MultiThread Semaphore version )
Bedirhan Bardakci - CSE331 Spring 2021 Assignment 2-a

Total Number of Printing Jobs To Simulate is 10

Free Buffer is 10 Kb

437 Kb , 449 Kb , 471 Kb , 350 Kb , 426 Kb , 251 Kb , 263 Kb , 289 Kb , 271 Kb , 245 Kb ,

Job 1 arrived with size 437 Kb
Job 1 cancelled and will not be queued, because it is bigger than printer max buffer size 10 Kb

Job 2 arrived with size 449 Kb
Job 2 cancelled and will not be queued, because it is bigger than printer max buffer size 10 Kb

Job 3 arrived with size 471 Kb
Job 3 cancelled and will not be queued, because it is bigger than printer max buffer size 10 Kb

Job 4 arrived with size 350 Kb
Job 4 cancelled and will not be queued, because it is bigger than printer max buffer size 10 Kb

Job 5 arrived with size 426 Kb
Job 5 cancelled and will not be queued, because it is bigger than printer max buffer size 10 Kb

Job 6 arrived with size 251 Kb
Job 6 cancelled and will not be queued, because it is bigger than printer max buffer size 10 Kb

Job 7 arrived with size 263 Kb
Job 7 cancelled and will not be queued, because it is bigger than printer max buffer size 10 Kb

Job 8 arrived with size 289 Kb
Job 8 cancelled and will not be queued, because it is bigger than printer max buffer size 10 Kb

Job 9 arrived with size 271 Kb
Job 9 cancelled and will not be queued, because it is bigger than printer max buffer size 10 Kb

Job 10 arrived with size 245 Kb
Job 10 cancelled and will not be queued, because it is bigger than printer max buffer size 10 Kb

Total Printed Pages : 0
```

10 File & 105 Kb Buffer

150 KB Max Buffer Size - Some File may be smaller than the Buffer so Total Print may > 0

```
Printer Spooler ( MultiThread Semaphore version )
Bedirhan Bardakci - CSE331 Spring 2021 Assignment 2-a

Total Number of Printing Jobs To Simulate is 10

Free Buffer is 150 Kb

388 Kb , 53 Kb , 375 Kb , 458 Kb , 392 Kb , 405 Kb , 206 Kb , 157 Kb , 435 Kb , 436 Kb ,

Job 1 arrived with size 388 Kb
Job 1 cancelled and will not be queued, because it is bigger than printer max buffer size 150 Kb

Job 2 arrived with size 53 Kb

Job 2 enqueued now with size 53 Kb, thread id 140054557722368, buffer left : 97 Kb

Job 2 starts printing for 0.53 seconds

Job 3 arrived with size 375 Kb
Job 3 cancelled and will not be queued, because it is bigger than printer max buffer size 150 Kb

Job 4 arrived with size 458 Kb
Job 4 cancelled and will not be queued, because it is bigger than printer max buffer size 150 Kb

Job 5 arrived with size 392 Kb
Job 5 cancelled and will not be queued, because it is bigger than printer max buffer size 150 Kb

Job 6 arrived with size 405 Kb
Job 6 cancelled and will not be queued, because it is bigger than printer max buffer size 150 Kb

Job 7 arrived with size 206 Kb
Job 7 cancelled and will not be queued, because it is bigger than printer max buffer size 150 Kb

Job 8 arrived with size 157 Kb
Job 8 cancelled and will not be queued, because it is bigger than printer max buffer size 150 Kb

Job 9 arrived with size 435 Kb
Job 9 cancelled and will not be queued, because it is bigger than printer max buffer size 150 Kb

Job 10 arrived with size 436 Kb
Job 10 cancelled and will not be queued, because it is bigger than printer max buffer size 150 Kb

Job 2 completed printing, Available Free Buffer : 150 Kb

Total Printed Pages : 1

...Program finished with exit code 0
Press ENTER to exit console.
```

10 File & 1000 Kb Buffer

1000 KB Max Budder Size (All File will be smaller than the Buffer so Total Print = 10)

```
Printer Spooler ( MultiThread Semaphore version )
Bedirhan Bardakci - CSE331 Spring 2021 Assignment 2-a

Total Number of Printing Jobs To Simulate is 10

Free Buffer is 1000 Kb

487 Kb , 124 Kb , 476 Kb , 156 Kb , 170 Kb , 170 Kb , 387 Kb , 364 Kb , 341 Kb , 86 Kb ,

Job 1 arrived with size 487 Kb

Job 1 enqueued now with size 487 Kb, thread id 140413851879168, buffer left : 513 Kb

Job 1 starts printing for 4.87 seconds

Job 2 arrived with size 124 Kb

Job 2 enqueued now with size 124 Kb, thread id 140413851879168, buffer left : 392 Kb

Job 3 arrived with size 476 Kb
Job 3 - File size : 476 Kb is waiting for available buffer, current empty buffer :417
Job 3 - File size : 476 Kb is waiting for available buffer, current empty buffer :442
Job 3 - File size : 476 Kb is waiting for available buffer, current empty buffer :467
Job 3 - File size : 476 Kb is waiting for available buffer, current empty buffer :492

Job 3 enqueued now with size 476 Kb, thread id 140413851879168, buffer left : 20 Kb

Job 4 arrived with size 156 Kb
Job 4 - File size : 156 Kb is waiting for available buffer, current empty buffer :45
Job 4 - File size : 156 Kb is waiting for available buffer, current empty buffer :70
Job 4 - File size : 156 Kb is waiting for available buffer, current empty buffer :95
Job 4 - File size : 156 Kb is waiting for available buffer, current empty buffer :119
Job 4 - File size : 156 Kb is waiting for available buffer, current empty buffer :144
Job 4 - File size : 156 Kb is waiting for available buffer, current empty buffer :169

Job 4 enqueued now with size 156 Kb, thread id 140413851879168, buffer left : 18 Kb

Job 5 arrived with size 170 Kb
Job 5 - File size : 170 Kb is waiting for available buffer, current empty buffer :43
Job 5 - File size : 170 Kb is waiting for available buffer, current empty buffer :67
Job 5 - File size : 170 Kb is waiting for available buffer, current empty buffer :92
Job 5 - File size : 170 Kb is waiting for available buffer, current empty buffer :117
Job 5 - File size : 170 Kb is waiting for available buffer, current empty buffer :142
Job 5 - File size : 170 Kb is waiting for available buffer, current empty buffer :166

Job 5 enqueued now with size 170 Kb, thread id 140413851879168, buffer left : 1 Kb

Job 6 arrived with size 170 Kb
Job 6 - File size : 170 Kb is waiting for available buffer, current empty buffer :26
Job 6 - File size : 170 Kb is waiting for available buffer, current empty buffer :51
Job 1 completed printing, Available Free Buffer : 74 Kb

Job 2 starts printing for 1.24 seconds
```

Job 2 starts printing for 1.24 seconds

Job 6 - File size : 170 Kb is waiting for available buffer, current empty buffer :76
Job 6 - File size : 170 Kb is waiting for available buffer, current empty buffer :100
Job 6 - File size : 170 Kb is waiting for available buffer, current empty buffer :125
Job 6 - File size : 170 Kb is waiting for available buffer, current empty buffer :150
Job 6 - File size : 170 Kb is waiting for available buffer, current empty buffer :175

Job 6 enqueued now with size 170 Kb, thread id 140413851879168, buffer left : 10 Kb

Job 7 arrived with size 387 Kb

Job 2 completed printing, Available Free Buffer : 28 Kb

Job 3 starts printing for 4.76 seconds

Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :35
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :59
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :84
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :109
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :134
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :158
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :183
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :208
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :233
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :257
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :282
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :307
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :331
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :356
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :381
Job 7 - File size : 387 Kb is waiting for available buffer, current empty buffer :406

Job 7 enqueued now with size 387 Kb, thread id 140413851879168, buffer left : 24 Kb

Job 8 arrived with size 364 Kb

Job 8 - File size : 364 Kb is waiting for available buffer, current empty buffer :48
Job 8 - File size : 364 Kb is waiting for available buffer, current empty buffer :73
Job 8 - File size : 364 Kb is waiting for available buffer, current empty buffer :98
Job 3 completed printing, Available Free Buffer : 117 Kb

Job 4 starts printing for 1.56 seconds

Job 8 - File size : 364 Kb is waiting for available buffer, current empty buffer :122
Job 8 - File size : 364 Kb is waiting for available buffer, current empty buffer :147
Job 8 - File size : 364 Kb is waiting for available buffer, current empty buffer :172
Job 8 - File size : 364 Kb is waiting for available buffer, current empty buffer :197
Job 8 - File size : 364 Kb is waiting for available buffer, current empty buffer :221
Job 8 - File size : 364 Kb is waiting for available buffer, current empty buffer :246
Job 8 - File size : 364 Kb is waiting for available buffer, current empty buffer :271
Job 4 completed printing, Available Free Buffer : 273 Kb

Job 5 starts printing for 1.70 seconds

Job 8 - File size : 364 Kb is waiting for available buffer, current empty buffer :296
Job 8 - File size : 364 Kb is waiting for available buffer, current empty buffer :320
Job 8 - File size : 364 Kb is waiting for available buffer, current empty buffer :345
Job 8 - File size : 364 Kb is waiting for available buffer, current empty buffer :370

Job 8 enqueued now with size 364 Kb, thread id 140413851879168, buffer left : 11 Kb

Job 9 arrived with size 341 Kb

Job 9 - File size : 341 Kb is waiting for available buffer, current empty buffer :36

Job 9 - File size : 341 Kb is waiting for available buffer, current empty buffer :60

Job 5 completed printing, Available Free Buffer : 79 Kb

Job 6 starts printing for 1.70 seconds

Job 9 - File size : 341 Kb is waiting for available buffer, current empty buffer :85

Job 9 - File size : 341 Kb is waiting for available buffer, current empty buffer :110

Job 9 - File size : 341 Kb is waiting for available buffer, current empty buffer :135

Job 9 - File size : 341 Kb is waiting for available buffer, current empty buffer :159

Job 9 - File size : 341 Kb is waiting for available buffer, current empty buffer :184

Job 9 - File size : 341 Kb is waiting for available buffer, current empty buffer :209

Job 9 - File size : 341 Kb is waiting for available buffer, current empty buffer :234

Job 6 completed printing, Available Free Buffer : 249 Kb

Job 7 starts printing for 3.87 seconds

Job 9 - File size : 341 Kb is waiting for available buffer, current empty buffer :258

Job 9 - File size : 341 Kb is waiting for available buffer, current empty buffer :283

Job 9 - File size : 341 Kb is waiting for available buffer, current empty buffer :308

Job 9 - File size : 341 Kb is waiting for available buffer, current empty buffer :333

Job 9 - File size : 341 Kb is waiting for available buffer, current empty buffer :357

Job 9 enqueued now with size 341 Kb, thread id 140413851879168, buffer left : 21 Kb

Job 10 arrived with size 86 Kb

Job 10 - File size : 86 Kb is waiting for available buffer, current empty buffer :46

Job 10 - File size : 86 Kb is waiting for available buffer, current empty buffer :71

Job 10 - File size : 86 Kb is waiting for available buffer, current empty buffer :95

Job 10 enqueued now with size 86 Kb, thread id 140413851879168, buffer left : 14 Kb

Job 7 completed printing, Available Free Buffer : 209 Kb

Job 8 starts printing for 3.64 seconds

Job 8 completed printing, Available Free Buffer : 573 Kb

Job 9 starts printing for 3.41 seconds

Job 9 completed printing, Available Free Buffer : 914 Kb

Job 10 starts printing for 0.86 seconds

Job 10 completed printing, Available Free Buffer : 1000 Kb

Total Printed Pages : 10

...Program finished with exit code 0

CONCLUSION

Even if it is known that processes are more heavy-weight than threads, and have a higher startup costs, in our sample we have just 2 process (1 fork operation) so it does not matter much for us. But in term of launching, project with MultiThread launches 1,5 seconds faster than Multi Process project

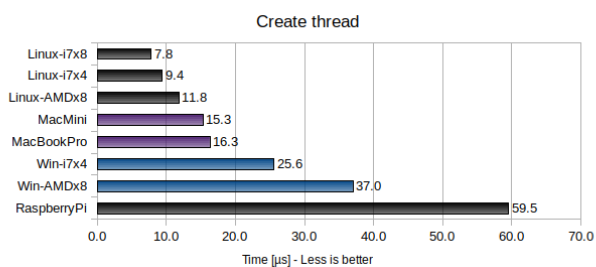
Results are near to each other for both projects

	Buffer Size (Kb)	Total File Count	Total File Size Printed (Kb)	Execution (seconds)
MultiThread	1000	10	2045	20,79
MultiThread	1000	20	4190	42,81
MultiThread	1000	30	6435	65,36
MultiThread	1000	100	24950	255.93
MultiProcess	1000	10	2045	20,81
MultiProcess	1000	20	4190	43,15
MultiProcess	1000	30	6435	65,61
MultiProcess	1000	100	24950	257.22
CPU : AMD Ryzen 9 390x				
Ram : 3600 Mhz				

Online compiler used : https://www.onlinegdb.com/online_c_compiler

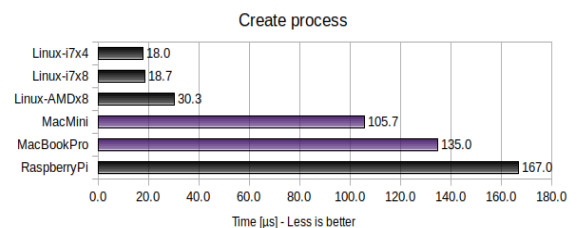
Creating threads

In this benchmark 100 threads are created. Each thread terminates immediately without doing any work, and the main thread waits for all child threads to terminate. The time it takes for a single thread to start and terminate is measured.



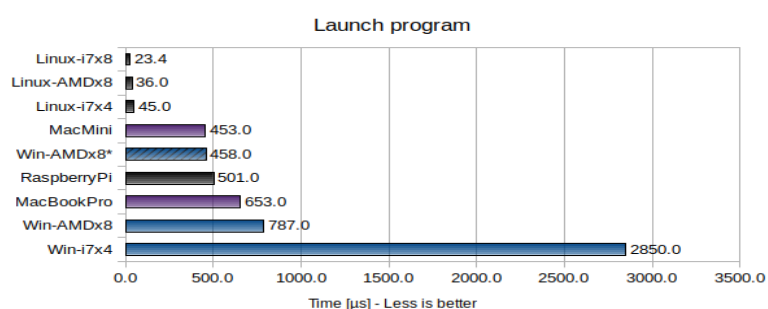
Creating processes

This benchmark is almost identical to the previous benchmark. However, here 100 child *processes* are created and terminated (using `fork()` and `waitpid()`). Unfortunately Windows does not have any corresponding functionality, so only Linux and macOS were benchmarked.



Launching programs

Launching a program is essentially an extension to process creation: in addition to creating a new process, a program is loaded and executed (the program consists of an empty `main()` function and exists immediately). On Linux and macOS this is done using `fork() + exec()`, and on Windows it is done using `CreateProcess()`.



REFERENCES

- <https://www.backblaze.com/blog/whats-the-diff-programs-processes-and-threads/#:~:text=Processes%20are%20typically%20independent%20of,threads%20in%20the%20same%20process.>
- <https://www.journaldev.com/36220/queue-in-c>
- <https://linuxquestions.org/questions/programming-9/can-someone-give-me-the-example-of-the-simplest-usage-of-nanosleep-in-c-4175429688/>
- <https://stackoverflow.com/a/63408673>
- <https://www.bitsnbites.eu/benchmarking-os-primitives/>
- <https://www.geeksforgeeks.org/using-fork-produce-1-parent-3-child-processes/>
- <https://stackoverflow.com/questions/807506/threads-vs-processes-in-linux>
- https://man7.org/linux/man-pages/man7/sem_overview.7.html