

Assignment 2

Parikshit	Vemula	Soumyadeep
Gehlaut	Chandrahaas	Das
220150009	Reddy	220010056
	220010062	

September 1, 2024

1 Introduction

This report presents the implementation of process management for pattern searching in a file, developed in three parts. The project demonstrates the use of process creation, inter-process communication, and termination techniques in a C++ environment.

2 Part I: Single Process Pattern Search

2.1 Objective

To search for a pattern within a specific portion of a file using a single process.

2.2 Implementation

- File: `part1_searcher.cpp` - The program searches a file segment and prints whether the pattern is found.

2.3 Results

- Output when the pattern is found: `[<pid>] found at <position>` - Output when the pattern is not found: `[<pid>] didn't find`

3 Part II: Multi-Process Pattern Search

3.1 Objective

To search for a pattern using multiple processes, each handling different chunks of the file.

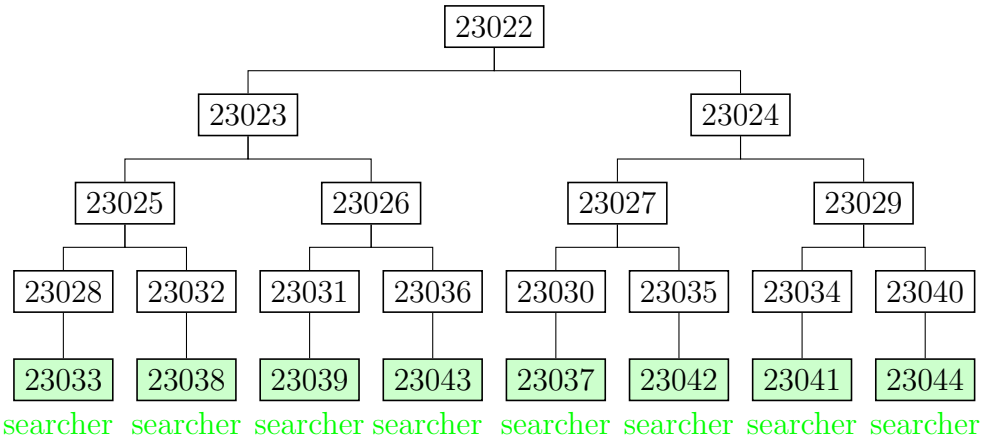
3.2 Implementation

- File: `part2_partitioner.cpp` - Each process forks additional child processes if the file segment exceeds a defined chunk size.

3.3 Results

The process tree generated during execution is shown in Figure below.

3.4 Tree diagram



4 Part III: Optimized Multi-Process Search with Early Termination

4.1 Objective

To optimize the search by terminating unnecessary processes once the pattern is found.

4.2 Implementation

- Files: `part3_searcher.cpp`, `part3_partitioner.cpp` - Processes send termination signals to other processes once the pattern is found.

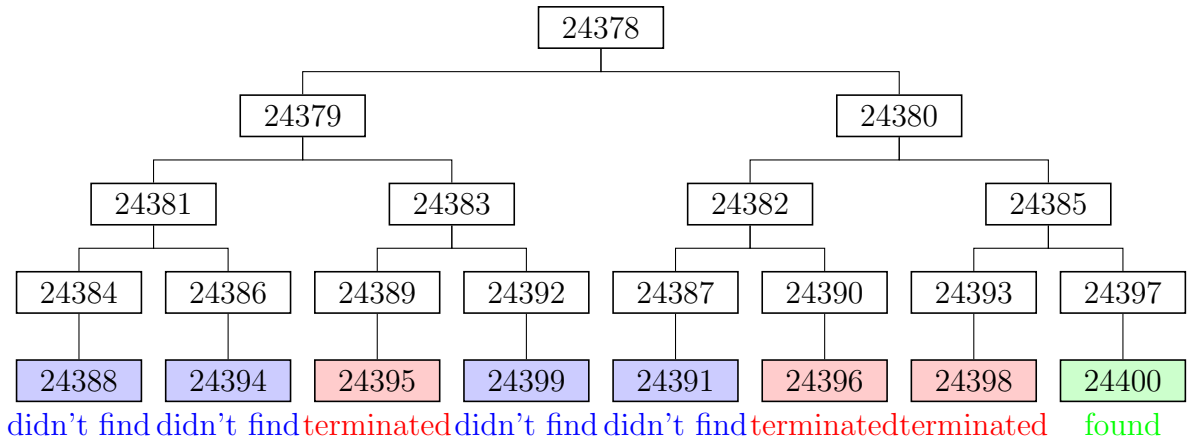
4.3 Results

The following output was observed:

```
./part3_partitioner.out file.txt NGTNIJGK 0 67108863 8388608
[24378] start position = 0 ; end position = 67108863
[24378] forked left child 24379
[24379] start position = 0 ; end position = 33554431
[24378] forked right child 24380
[24380] start position = 33554432 ; end position = 67108863
[24379] forked left child 24381
[24381] start position = 0 ; end position = 16777215
[24380] forked left child 24382
[24379] forked right child 24383
[24381] forked left child 24384
[24382] start position = 33554432 ; end position = 50331647
[24383] start position = 16777216 ; end position = 33554431
[24384] start position = 0 ; end position = 8388607
[24380] forked right child 24385
[24381] forked right child 24386
[24382] forked left child 24387
[24384] forked searcher child 24388
[24385] start position = 50331648 ; end position = 67108863
[24386] start position = 8388608 ; end position = 16777215
[24387] start position = 33554432 ; end position = 41943039
[24383] forked left child 24389
[24389] start position = 16777216 ; end position = 25165823
[24382] forked right child 24390
[24390] start position = 41943040 ; end position = 50331647
[24387] forked searcher child 24391
[24383] forked right child 24392
[24385] forked left child 24393
[24386] forked searcher child 24394
```

[24389] forked searcher child 24395
[24392] start position = 25165824 ; end position = 33554431
[24393] start position = 50331648 ; end position = 58720255
[24390] forked searcher child 24396
[24385] forked right child 24397
[24393] forked searcher child 24398
[24392] forked searcher child 24399
[24397] start position = 58720256 ; end position = 67108863
[24397] forked searcher child 24400
[24394] didn't find
[24386] searcher child returned
[24399] didn't find
[24392] searcher child returned
[24391] didn't find
[24387] searcher child returned
[24388] didn't find
[24384] searcher child returned
[24400] found at 64520807
[24395] received SIGTERM
[24398] received SIGTERM
[24396] received SIGTERM
[24400] received SIGTERM
[24385] received SIGTERM
[24390] received SIGTERM
[24393] received SIGTERM
[24397] received SIGTERM
[24383] received SIGTERM
[24380] received SIGTERM
[24389] received SIGTERM
[24378] received SIGTERM
[24382] received SIGTERM
[24379] received SIGTERM

4.4 Tree diagram



5 Conclusion

This project demonstrated effective use of process management in pattern searching. The implementation of early termination significantly improved efficiency by avoiding redundant searches.