The UNIX Fork Command

Use and Application of fork() within the UNIX Operating System

Emmet Allen
New York University Tandon Bridge Program
New York University
New York City, New York
ea2609@nyu.edu

***Abstract*—The Unix Fork command is a command that has helped manage the memory of processes by the Operating System. This is done by making a clone of a current process and creating a newly duplicated process known as the child process, while still using parent processes memory. The child process has the same data that would be found in the now parent process, from which that child could further create more child processes.**

***Index Terms*—Fork, Memory Management, Operating Systems, Paging, Processes, UNIX,**

I. Introduction

The Unix Fork command is an operating system level command in which a process is duplicated from a parent process to a child process, creating two duplicate processes with the same data necessary for the new child process to run. [1] Though before going further into how the fork command is used within the operating system, we need to understand the underlying memory management mechanisms that are used for this command to take place.

II. Paging and Memory Management in Operating Systems

A. Paging

The memory management mechanism known as "paging", is used in most modern operating systems, including the Unix family of operating systems. Paging, or in this case simple paging, is used to break down main memory from a large concurrent memory block to substantially smaller equal blocks of memory called "frames" [2]. The operating system will then create a new process, and while this process is in the "new" state, will divide the process into more manageable sizes called "frames", where any one "frame" is equivalent or smaller than one "page" of memory in main memory.

B. Operating Systems use of Paging

The operating system will determine which frame of memory will hold what page of a process and will record the "frame number" of any page that is a part of the current process within its "Process Control Block", using a mapping table called the "Page Map Table" [2]. The use of paging helps the Operating System use the minimal amount of main memory, for a process without losing substantial amounts of memory due to internal fragmentation, and the operating system can access each page in constant time, due to the properties of the main memory being random access.

III. Fork command and Process Management

What the Unix Fork command does while creating a child process, from a parent process, is not only duplicate the code which is used by the parent process, but it also copies the same Page Map Table, that was used by the parent process, and recorded by the operating system in the process control block [1]. This is because when the fork command is called, it is not creating a new copy of the process, as much as it is creating an exact duplicate of the process in which the operating system can use the same Page Map Table that is being used by the parent process so that the child process can use those same pages to perform the various task of the process [1]. This results in both of the processes being marked as "Copy On Write" so that both the parent and child processes have access to the same memory resource or frame.

However, if the child process or the parent process were to change tasks in a certain way, the operating system would not need to replace that entire frame of memory, and instead would copy the memory of that page whenever those changes are made, and then write them to that particular instance of memory or frame [1].

IV. Conclusion

The Unix Fork command has many applicable uses when it comes to the creation of processes while optimizing memory management. With the operating system's use of paging to minimize as much main memory fragmentation as possible, and having Fork create processes that use the same memory as the parent processes it creates, forking is an undeniable way to save memory, while having multiple processes.

References

[1] Free Software Foundation, "Linux Manual Page: Fork(2)", 2021, [Online] Available:https://man7.org/linux/man-pages/man2/fork.2.html [Accessed June. 1, 2021]

[2] A. S. Tanenbaum., H.J. Bos, "Modern Operating Systems," 4th ed., Pearson Higher Education, 2015, pp.181–254

Identify applicable funding agency here. If none, delete this.