Byron Laferriere

Professor Hardikar

IT-365

Milestone Four

* *How the chosen operating system approaches process management*

The Linux operating system offers user-controlled process management through the command line, through a series of commands, that can help the system perform better when properly used. When closing processes, depending on the demand of the process on the CPU, they can continue running in the background and require the user to step in and manage the process closure personally (Sanchez, 2016). The Linux system has four commands, with two variants, that help a user quickly and neatly manage processes running on the system. By using the commands ‘top’, ‘htop’, ‘ps’, ‘pstree’, ‘who’, and ‘kill’, the user is provided with powerful pipelining capabilities to conduct process management on the Linux system (Sanchez, 2016). Every process can also be run in two separate ways on the Linux operating system, the foreground and background processes. The option to launch a process in the foreground receives input from the command line (stdin) and outputs to the screen (stdout) (Unknown, 2021). No other processes can run on top of this foreground process on the same terminal until the process is killed (Unknown, 2021). Also, any foreground process can be turned into a background process by adding a ‘&’ to the process command. Background processes run on their own until they receive input from the command line and allow for other processes to run in the foreground simultaneously (Unknown, 2021). There are five types of processes in the Linux system that can be run under the background/foreground format which are: parent process, child process, orphan process, zombie process, and daemon process (Unknown, 2021). Child processes are created by parent processes and cannot be altered unless going through the parent process. Orphan’s are child processes that were not closed when the parent process was closed and will run until independently closed. Zombie’s are processes that were closed but are still running and consuming CPU usage. Lastly, daemon processes are just processes needed by the system that run in the background.

* *How the chosen operating system approaches memory management*

This section on how Linux manages its memory could go on and on, however, I will attempt to summarize how Linux uses virtual memory to help support its physical memory. Using virtual memory provides a nearly endless additional memory by creating large address spaces to be used by the system and making the system memory seem much larger than it is (Dev. Team, 2019). It also helps to provide additional protection to the system by providing each process its own virtual address space (Dev. Team, 2019). This practice isolates processes and does not allow them to affect other processes running on the system, while using virtual memory hardware to protect against unwanted code writing by rogue applications (Dev. Team, 2019). Linux also uses memory mapping and fair physical memory allocation to balance process memory demands and quickly access addresses through the maps. There is also a shared virtual memory form of Inter Process Communication (IPC) that is supported by Linux, which is the Unix System V shared memory IPC (Dev. Team, 2019). This allows processes to exchange via a common memory source that can be accessed by all of them. One of the most interesting approaches to memory management was Demand Paging. After just recently learning about the importance of paging for large page tables, it was interesting to read about how Linux approaches the issue. Demand paging saves physical memory by only loading virtual pages that are currently be used by executing programs (Dev. Team, 2019). This produces accurate results in a quicker manner by only loading what is necessary for each page. Linux uses four types of memory-management related caches which are: buffer caches, page caches, swap caches, and hardware caches. The one drawback to using these caches is that they will cause a system crash if one of the caches becomes corrupted (Dev. Team, 2019).

* *How the chosen operating system approaches file management*

The Linux/UNIX operating system has, in my opinion, a superior file management process that is available through the command terminal. File management can be completed through the terminal without ever leaving the window. Using commands like ‘nano’ allow for file creation and writing to the file, all within the command terminal. Users can also use ‘touch’ and create thousands of files at once through the pipelining capabilities available in the command terminal. The Linux file system follows the tree-like structure using files and directories to contain those files (@manav014, 2020). Directories can also be easily created through the command terminal by typing ‘mkdir’. Users can also move files using ‘mv’ or remove unwanted files using ‘rm’. There are many commands that can be used for file manipulation of all sorts on the Linux operating system and once they are learned, they make file management easy and quicker to perform compared to a system like Windows. There is a Virtual File System (VFS) present in Linux that recognizes four types of main objects: an inode object, a file object, a superblock object, and a dentry object. The regular file type is used to store items like readable files, binary files, image files, compressed files, etc… (Anne, 2016). The directory file type is used to store files, folders, or special files in containers on a physical device (Anne, 2016). Combining these simple, yet powerful file combinations is just one reason why this FHS system is superior in file management strategies.

* *How the chosen operating system approaches system resource management*

Linux provides resource management through the usage of Control Groups, subsystems, and tunable parameters (Prpic et al, 2011). Control groups are useful in resource management because it limits tasks to a single control group in any single hierarchy (Prpic et al, 2011). This provides foundational structuring and limits the tasks from consuming unwanted resources. Control groups also allow for subsystem grouping so they affect all tasks in a single hierarchy (Prpic et al, 2011). Hierarchies can be easily deleted and created, or even reassigned through using this system of control groups. Lastly, control groups benefit the resource management process on Linux due to their design allowing for highly specific configurations (Prpic et al, 2011). Each process on the system could be a member of each hierarchy, which all have a single attached subsystem. This configuration allows for an administrator to have absolute control over every defined parameter for each task (Prpic et al, 2011).

References:

@manav014. (2020, February 24). File management in Linux. Retrieved April 05, 2021, from <https://www.geeksforgeeks.org/file-management-in-linux/>

Anne, S. (2016, February 10). File types in Linux/Unix explained in detail. Retrieved March 14, 2021, from <https://www.linux.com/training-tutorials/file-types-linuxunix-explained-detail/>

Developer Team. (2019, Summer). How Linux OS memory management works. Retrieved April 05, 2021, from <https://blockchain.dcwebmakers.com/blog/how-linux-operating-system-memory-management-works.html>

Prpic, M., Landmann, R., & Silas, D. (2011). Resource Management Guide - Managing system resources on Red Hat Enterprise Linux 6. Retrieved April 05, 2021, from <https://linux.oracle.com/documentation/EL6/Red_Hat_Enterprise_Linux-6-Resource_Management_Guide-en-US.pdf>

Purnaye, S. (2018, March 01). Linux memory management. Retrieved April 01, 2021, from <https://superchargedcomputing.com/2018/03/01/linux-memory-management/#:~:text=Linux%20Memory%20Management%201%20Memory%20Address.%20This%20address,in%20Linux%2010%20Data%20structures%20for%20segmentation.%20>

Sanchez, R. (2016, February 10). Linux process MANAGEMENT: Commands you should know. Retrieved April 01, 2021, from <https://www.hostingadvice.com/blog/linux-process-management-commands-know/#:~:text=Linux%20Process%20Management%3A%20Commands%20You%20Should%20Know%201,pstree.%20...%205%20who.%20...%206%20kill.%20>

Unknown. (2021, March 03). Linux process MANAGEMENT: Five types of process in Linux management. Retrieved April 01, 2021, from <https://www.educba.com/linux-process-management/>