**Review On The Paper Of “On The Duality Of Operation System Structure”**

Yukui Ye

SUID: 439644268

This paper construct two abstract models for observing properties on operation system design, which is “Message-Oriented System”(MOS) and “Procedure-Oriented System”(POS). MOS is more likely like a system that encoded its application into data and associated processes with system resources, which would lead to a relatively static environment. But POS is implemented in the opposite way, it encoded its system resources into common or global data structures which enables shared memory and shared data mechanism, and the applications on POS are associated with processes, this special mechanism requires a relative dynamic environment between processes and applications to do synchronization. The author compare and map the similarity of program logic, code and performance within the two system by identifying properties in the concepts of process, synchronization, and interprocess communication occur within these system. The paper indicate that the two systems are duals of each other since their constructs have direct counterparts, programs are logically equivalent, and the performance is identical to each other, no inherent differences between these two system, then there comes a question about how to choose operation system design between these two system since they are relatively similar to each other. Finally the author do give us a good answer.

Specifically , the author separately dig into these two different operation system design. MOS operate on relatively small and static processes, and these processes executes in static context. MOS can easily passing message through message channel, it creats queues for processes and these processes are given operations for sending a message, waiting for any messages or a particular type of message, and the pre-emption happened when message arrive at a higher priority process. While POS operates on a relatively large rapidly changing processes, Cooperation between processes can be done through locked data structures, and the pre-emption happened when a higher priority get a lock it was waiting on.

I think the duality mapping between these two system is the foundation of this paper. In MOS, creat process can be mapped in new/start in POS; message channels,message,ports can be mapped with procedure identifier in POS; SendMessage, AwaitReply(Immediately) can be mapped in Procedure Call; SendMessage, AwaitReply(delayed) can be mapped in FORK,JOIN in POS; SendReply can be mapped in return(from procedure); Main loop of standard resource manager,WaitForMessage,case statement can be mapped in lock,entry attribute in POS; Arms of the case statement selective mapped with entry; Waiting for message mapped with condition variable, wait, signal in POS. All programs can be transformed using the duality mapping showed in the paper, such transformation has no effection on the logic of the program, such the execution speed would not change, the amount of information and code, the interaction between programs, as well as the total lifetime computation, are the same as before. but the constriction is that this works only if the model employed in the paper has been strictly used. The two model is similar in execution time, computational load, queuing, wait times and execution time of the system primitives.

Therefore, each system has its own merits in structure, lock of one system has directly counterpart in the other. We can choose model based on machine architecture and programming environment. Once choice has been made between these two system, many good properties for a good system comes naturally. To make a better, more consistent, more reliable system at lower cost, we can eliminate degrees of freedom in design process. The author also make a postulation about devise a uniform way of modeling the interactions between system components.