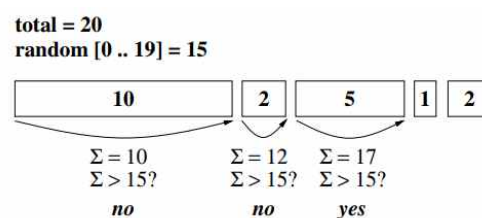


Lottery Scheduling: Flexible Proportional-Share Resource Management, OSDI 1994

(1)

It is preemptive scheduler for fair resource allocation (time slice in this case).

For each turn, it selects random target ticket number (15 in this case) for winning ticket and allocate tickets (ranging 0~19) to each clients who is ready. Each client's get a number of tickets which are proportional to their own weight. The client who get the ticket with target number will get the resource (time slice in this case). The more ticket you have, the more chance for resource you will get.



(2)

strong - prevent starvation by giving every clients opportunity to get the resources (even the clients with small weight).

weak - unexpectancy: there is no regularity in choosing which client to get the resources. Therefore, cannot expect which clients to get the resource next (time slice in this case).

It will satisfy fairness in the long run, but it cannot guarantee fairness in the short run. With high amount of tickets the randomness of this policy grows.

(3)

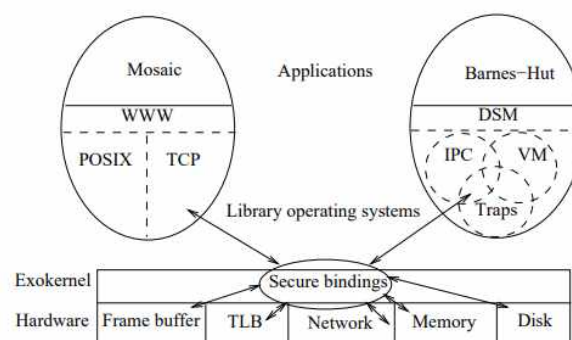
Because it uses randomness of lottery, it cannot guarantee anything. I suggest to set epoch (a unit period after which fairness guaranteed) and make a barrier end of each epoch. to keep client with low weight from taking resource too much.

Exokernel An Operating System Architecture for Application-Level Resource Management, SOSP 1995

(1)

Even when application running on a OS does not need it, OS has a lot of abstractions, interfaces and protections for genericity. It can hinder the application to get flexibility on running them and utilize all the resources which can be used in other way.

By adopting library operating system, exokernel give applications chance to set and run the essential and minimum function of OS they really want. By doing this, applications even can directly manage hw resources and it can improve hw utilization.



(2)

Strong – Can get optimized performance by not having undesirable abstractions or preventions.

Weak – Does not used in major commercialize OS. Application developer should know how to apply the OS libraries what resource they want. Even if they know how to do it, it takes a lot of time.

If it does not have many cooperative maintainers like linux or every vendor provide proper libraries for their products, it will be super hard to build and keep the OS libraries.

(3)

Exokernel provide really basic functions for binding those Library OS, and let Library OSs do things for applications. I think it is quite similar to the concept of hypervisor. I think it would be possible to blinding the boarder line between VMs and Applications with lib OSs.