OS HOMEWORK
WEEK-6
TANU SHREE

Q.1. The question says about the situation when a process creates huge no. of threads. Since each thread gets equal chance to be run, the process with huge no. of threads clogs the CPU, making other processes with few threads to starve.

To solve this problem, CPU can be divided into smaller groups called control group. Control groups are assigned to processes and each process, irrespective of the no. of threads will only have access to the group it is assigned to. This way the processes will not impact other processes.

Q 2. Many to many multithread model means, multiple user threads are mapped to same or fewer number of kernel threads.

We know that the scheduler assigns the cores to kernel threads, not to user threads.

a.) When the no. of kernel threads allocated to user threads are less than the no. of processing cores, not all processing cores are utilized. Only the cores assigned to kernel threads are used, rest are idle. Since, cores are not assigned to user threads, even though there are large no. of user threads, they can't be assigned.

b.) No. of kernel threads = No. of processing cores

All the processing cores are assigned to all the kernel threads, so all of them are utilized at first But, if any kernel thread is blocked, in that case the processor assigned to that kernel will become idle. Therefore, the processing cores may or maynot be fully utilized.

c.) No. of kernel threads > No. of processing cores.

In this case, the performance will be better than previous case. First, all the cores will be fully utilized, if, any kernel thread gets blocked, instead of becoming idle, core can be assigned to other kernel thread to start execution.

Q.3 Scheduler activations can be considered as kernel-level thread that provide an additional API to the ULT library. In this, the kernel would provide information to ULT by providing a no. of upcalls. These would indicate to the application that a kernel thread has been blocked, or unblocked, that a processor has been preempted or that a processor has been added.

But these require additional API and include several upcalls. Therefore, this requires several changes and hence not portable.

8.4. Advantage

The advantage of progressive round robin is efficiency. If the quantum is fixed, interrupts are generated with short intervals, resulting in overhead because of context switches. Progressive round robin avoids frequent context switches and the efficiency of CPU increases.

Disadvantage

Progressive round robin will slowly result in larger quantum time. This will reduce the overhead of context switch, but the response time will be affected. The short jobs will have to wait longer on the ready queue to be executed. This might also lead to starvation sometime.

Also, if the quantum time of a process progresses to have too large time, other more such processes in addition to that will slow down the CPU.

Q.5 Given, RR scheduling
time quantum = q
context switch time = s ms
avg thread run time = t ms.

a.) CPU effeciency = $\frac{\text{useful CPU time}}{\text{total CPU time}}$

∴ CPU wastage = 1 − CPU efficiency

i.) $t < q$: When $t < q$, the process completes before quantum time.

∴ CPU efficiency = $\frac{t}{t+s}$

Process took t CPU time to run completely and s time of CPU was taken for context switch.

∴ CPU wastage = $1 - \frac{t}{s+t} = \frac{s+t-t}{t+s}$

$= \frac{s}{t+s}$

ii.) $t >> q$

When, t is much larger than quantum time, total no of context switches for the process would be t/q.

∴ Total context switch overhead = $\frac{s \times t}{q}$

Process take total of t time of CPU to run completely.

∴ CPU efficiency = $\frac{t}{t + st/q} = \frac{q}{q+s}$

& CPU wastage = $1 - \frac{q}{q+s} = \frac{q+s-q}{q+s} = \frac{s}{q+s}$

iii.) When q approaches 0, there will be frequent context switches. In fact all of the CPU time will be busy in context switch.

∴ The CPU wastage is 100%.

(b.) For wasted CPU time to be 50%, effciency is also 50%.

$$\therefore \text{CPU effeciency} = \frac{\text{CPU time useful}}{\text{CPU time useful + overhead}}$$

Suppose, the process completes in quantum time, then, $t = q$.

$$\therefore \text{efficiency} = \frac{q}{q+s}$$

$$\therefore \text{waste} = 1 - \frac{q}{q+s} = \frac{s}{s+q}$$

which is, 50%.

$$\therefore \frac{s}{s+q} = \frac{1}{2} \Rightarrow 2s = sq$$

$$\Rightarrow s = q.$$