Sahil Jindel, 18CS10048

Date...../...../.....

Page.....

OS CS39001
Take Home Assignment 1

(1) (a) Should not be privileged since user program could rather use a syscall to put CPU in privileged mode.

(b) Should be privileged in order to protect memory from corruption.

(c) Should not be privileged as value at a CPU register could be changed by a user program without kernel intervention. However few registers may require privilege to be accessed (like location to IDT).

(d) must be privileged as otherwise some process could take over the control of CPU by disabling the interrupt system.

(e) Not privileged since the user program is only reading the status of the device and won't need to use that I/o device.

(2) In order to avoid the switching overhead between different syscalls caused by the various request, we can pool them together in a single syscall, thus making the execution faster.

Sahil Jindal 18CS1C0048

(3) (a) This will help in improving performance of processes that depend on main memory and CPU bound processes as the CPU with twice speed means doubling clock frequency which allows almost double parallel threads to run. However no significant effect will be seen on I/O bound processes.

(b) More memory provides space for storing the process states and help processor to handle more processes by allowing it to load & switch the process states without page faults. However, no significant improvement in 'm'.

(c) No effect on CPU bound processes will be seen. However I/O bound processes will be made faster and hence "m" would increase if a lot of I/O bound processes are present.

(d) As explained in (a), CPU with twice speed will help in doubling the no. of processes that could be run. Also, doubling main memory would give more spare for storing process states. Combined, it will improve the efficiency effectively.

Sahil Jindal 18CS10048

Date...../...../.....

Page.....

(4) At each fork, no of processes doubles as a child is created.
Hence no of processes $= 2^4 = 16$ processes

(5) System call is a deliberate interrupt by a user process to request some kernel function whereas an exception is an interrupt that occurs when an error or illegal action is requested.
System call is synchronous whereas exceptions are not since they occur only on an illegal action.
Example of system call: Accessing files, I/O process
Example of exception: Division by zero.

(6) We know that
(a) total cost = cost of service time + cost of waiting time per user

$$= \frac{S}{NT} + \frac{WN}{M}$$

[Since cost of service time $= \frac{(S/T)}{N}$ and cost of waiting time per user $= \left(\frac{W}{M}\right) * N$.]

→ Total time $= \frac{S}{NT} + \frac{WN}{M}$

For max total time, diff wrt N,

$$0 = -\frac{S}{TN^2} + \frac{W}{M} \Rightarrow N = \sqrt{\frac{MS}{WT}}$$

Hence proved.

Sahil Jindal 18CS10048

Date...../...../.....

Page.....

(b) Putting the values,

$$50 = \sqrt{\left(\frac{5}{1}\right)\left(\frac{200\$/hr}{W}\right)}$$

$$\Rightarrow 2500 = \frac{1000}{W} \$/hr$$

$$\Rightarrow W = 0.4 \$/hr.$$

Ans7 (a) Ready to Blocked: Occurs after the process starts running and gets to blocked mode due to events like I/O waiting.

(b) Ready to Ready Swapped / Block to Blocked Swap

When system is low on primary memory and unable to hold all the active processes, the states are shifted to secondary memory.

(c) Ready Swapped to Ready / Blocked Swap to Block

Occurs when the system scheduler daemon selects the process to be read back into primary memory.

(d) Block to Ready / Blocked swap to Ready swap

When the event that interrupted a process gets over, it is transitioned back to ready state. Eg I/O waiting ends.

Sahil Jindal 18CS10048

Page.....

Ans 8 The entries are :-

1. Process State
2. Process Number
3. Program Counter
4. Registers
5. List of open files
6. CPU scheduling information
7. Memory management information
8. I/O status information
9. Accounting information
10. Location of the process control block.

~~(a) Can store information about the change state of registers, i.e. to know whether they are changed~~

(a) Can store information about the occupancy of registers in order to know if they are occupied or freed. By storing this information, we can choose not to update the unchanged registers, thus reducing context switch overhead.

(b) Can fasten the loading/storing of PCB by giving special registers for PCB only which could store PCB into the CPUs register itself.

Date...../...../.....

Page.....

Sahil Jindal 18CS10048

Ans9 CPU bound processes spends most of the time on executing code, thus an example of CPU bound process is compression/decompression of files.

I/O bound process spend time submitting I/O requests. Thus an example of I/O bound process can be a shell waiting for input from the user.

Ans10 Short term schedulers:

(a) Also called CPU scheduler, it ensures which program is suitable for processing.

(b) It is minimal in time sharing systems.

(c) Speed is fastest.

Medium term schedulers

(a) Also called process swapping scheduler, it removes the processes from the memory.

(b) It is part of time sharing systems.

(c) ~~correct~~ ff. Faster than long term but slower than short term scheduler.

Sahil Jindal 18CS10048

Page.....

Long term ~~slod~~ scheduler

(a) Also called job scheduler, it regulates the programs which are selected for processing.

(b) Absent in time sharing systems.

(c) Slowest among these three schedulers.

Ans 12 Since kernel needs these instructions to be loaded during boot time, it requires their location. Thus fixing location of these instruction helps kernel.

Ans 13 (a) Using dynamically allocated arrays

Pro: Fast access to PCB index and parent process.

Con: Altering size of the array makes insertion/deletion slow.

(b) Single linked list

Pro: Insertion/Deletion is faster.

Con: Access index of PCB and its parent is slow.

(c) Double linked list

Pro: Faster access to parent compared to single linked list. Insertion/Deletion fast

Con: More usage of memory to store parent pointer