

Suppose, time quantum = 50 ms -> counting is done using CPU clock -> done thru hardware, intersupt the OS every 50 ms a process might not utilize all of the time quantum But \rightarrow process ends earlier CPU does $\langle \rightarrow interrupts$ due to T/0not wait, errors, etc. Switches - program might wait for p2 to p3 input switching overhead \rightarrow if switching overhead is ÌM → (save program counter, etc. comparable to time quantum program control block of P2) then robin round algorithm \rightarrow (restore ... of ρ_2) is useless



-> Robin Tound algorithm requires hardware support (unlike the ones counter, clock, etc. for Batch OS) Priority Scheduling Ready { Processes in ready state -> each process in ready state has Q/queile P1/3 P2/4 P3/1 priority. a La range of numbers/integer picked first -> no agreed, universal sange \rightarrow ordered based on their priority. among designers : 0 —> highest 1 priority \rightarrow good way to reorder \rightarrow heapify. 0 1023 -> least 7 -> thow are priorities determined ? priority



	degrad	le H	he	priorit	y of	a	pro	cess	if	ìt	starts	usi	ng r	nose		
	CPU	time	2				l		•				U			
07	Nov	202	h													
OS		CPU	- Sched	uler												
		→ alk	a (lispate	her	(disp	atches	jobs)							
				l				0								
	Run	mine	ot a	DTOC	ess 7-	→ a	ppe <i>o</i> rs	Sim	ultane	ous						
		PUL	Schodu	line			<u>ل</u> ه	but not								
				0	V											

Modes of execution ______ to seperate user's and OS area → 1 1) User mode --> user's program running -) 2) System mode / control mode / Supervisor mode / kernel mode / previledged mode → OS prog8am -> previledged instructions -> executed by the OS specific opcode ---- user's program cannot execute this instructions (interrupt, illegal instruction) tlow to distinguish b/w user and OS? (CPU -> only one) that denotes a mode bìt -> one







	CLASSA	inste	
	Page		
	user mode Moders		
	de =1, emit	allow to commission? (pass on info)	
	Sys Moth Medico	5 @ How to present have conditioned	
		(the new posens herance.	
	program (uterform MPE)	100 process chick hisson	
	and the second	2 than to receiving dependences	
	capet OS,	M fr hude look p frap." Har h make p with for p, i data	
		p brredy's	
	And the fait of the		
	2	CRUSchuling CRU seld, run Manage	
		OS Filigun critical	
		Autoral B/	
		renditive OS (down by OS)	
		his to suching some part frond to user	
		Tentres Tentes	
		Lipe kund OS	
	the back	Po access files as through knowl yadda yadda	
		par overhed	-
		Kend OS Moren	
		A CONTRACT OF THE OWNER OWNE	
	1		
			-











- \rightarrow Concurrency \rightarrow allow all tasks to make progress.
- \rightarrow Parallelism \rightarrow perform more than one task simultaneously.
- -> Concurrency can exist without parallelism
 - -> CPU schedulers were designed to provide the illusion of parallelism
 - by rapidly switching processes
 - ★ Programming challenges
 → Identifying tasks
 Should be independent of one another
 - -> Balance: separate thread for a very simple task may not be worth the cost
 - -> Data splitting -> Data dependency -> testing and debugging



