				The second secon	
	Alloc Anglindone wat out our contrasts which can	TO A to Associate to the Associate agreement with the Associate As			A
				(PU schedoling alle in-terms at	Aim: Sad & i
				6 3	79
: 16				algoritan & on a maintiful time	
20 R				274	af Pound T
- Jnayer		3		Ht performance	Pabin (RR)
ST. W				Theory:	Ain
ChA OF	lime Gunchum i protess.	the rest	Had Kenute Bud Round	1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	-
		1 705 0			β 📂
allocated for a unit	1 1 1 1	d Breeze	tracess scheduling that determines executed by the used scheduling a Round Rabin Sche	duling of C	1 1 1
Stated time quartum  the queue The queue The college of Engineering & 1	A fixed to	traphyce tied time at quant	He sa	testand & imple duling Algorithm of mairing fime scheduling in appearance of the control of the	ingleme
Har quarter , it is a	A fixed time slive a	Scheduling Quantum The hime au A 1961ic r	the order (PU one algorithms is	duling Algorithm & of waiting fine & & & & & & & & & & & & & & & & & & &	inglementation
ا م <b>ر</b> ۾	A fixed time slive assigned to		the order in the CPU one of algorithm's in B	testand & imple duling Algorithm of mairing fime scheduling in appearance of the control of the	inglementation at

1	air Scheduling: Ensure that all processes get equal			
	share of time slice.			
	flicient for Time sharing systems: Frequently used			
	n multi-user & interactive systems.			
1	Key Parameters:			
	Arrival Time (AT): Time at which a process arrives			
	Burst Time (BT): Total execution time for which a			
	process can run before terminating			
	Time Quantum: (TQ): The fixed time for which a			
	process can run before switching.			
	Maiting Time (WT): Total time a process spends			
	lugiting In the ready queue.			
	Tumaround Time (TAT): Total time taken from			
	axrival to completion			
Formulas	De front of love for the hand hand to the firm			
	Turnaround Time (TAT) = Completion time (CT) -			
	Amiya) Time (AT)			
	Waiting Time (WT) = Turnamend Time (TAT) -			
	Burst Time (BT).			
	*			
ST. VIN	CENT PALLOTTI COLLEGE OF ENGINEERING & TECHNOLOGY, NAGPUR - 441 108			

Algorithm:	START:
i	Input the number of processes
11/	Input the (AT) & (BT) for each process
— ḿX	Set a fixed time quantum
iv >	Place process in a queue based on (AT)
٧)	Execute each process for the given time quantum
	a move it to the end of queue if it is not
	Completed.
<u> </u>	Confinue to until all processos dere completed
Viik	Compute (TAT) & (NT)
	Compute Average (AT) & Average (WT).
<u>                                      </u>	Display Kesults
<u> </u>	STOP
C1 cost	
	#Include < iostram>
Implementa.	#include <queque< td=""></queque<>
	using namespace sta
	De la
	Struct Process ?
	int id, arrival, burst, remaining,
	Completion, tumaround, marting;
	vaid mund Robin Projess projesces [] int n
	int timeQuantum) {
	Iqueve Lint> 9:
	int time = 0. completed = 0.
ST. VIN	CENT PALLOTTI COLLEGE OF ENGINEERING & TECHNOLOGY, NAGPUR - 441 108

for (int i=0; i <n; i++)="" td="" {<=""></n;>
processes [i]. remaining = processes [i].
burst;
for (int i=0; i <n; i++)="" td="" {<=""></n;>
if [processes [i]. qmival===0) {
q. push (i);
}
While (!q.empty ()) {
int index = q.font();
9.pop & ( ) :
if (processes [index]. remaining > time Quantum) {
time to time Quantum.
Processes Findex T. remaining -= time Quantum;
}else {
time += processes [index]-remaining;
processes [Indextormaining = 0.
processes Findex J. completion = ce time
Completed ++:
for (int =0; i <n; i+t)="" td="" {="" {<=""></n;>
if [processes [i]. arrival = time && processes [i]
(memaloling 70 && find (q.begin 1) quendl)
i)==q.ind()) {
q.push().
ST. VINCENT PALLOTTI COLLEGE OF ENGINEERING & TECHNOLOGY, NAGPUR - 441 108

	}
	H [moreses [1]. arrival ( lime
	if (process [index]. remaining 10)}
(10)	9. push (index):
	} -
	for (int 1 = 0; i < n; i++) } "
	processes [i]. turnaround = processes [i]. completion
	- processes [i] amiral;
	processes [i]. waiting = processes [i]. tumoround
	- projesses [i] burst:
the state of the state of	Cout << "PID Arrival burst completion Turn around time
15.1	writing Time 'In '):
19	tor (int 1=0: 12n; i++) {
<u> </u>	cout << provesses [i].id << "  t" << provesses [i]
	"It" > tered. [i] mossion >> "H" >> lawina
*	<pre><pre>&lt;<pre>processes [i]. completion &lt;&lt; "It" &lt;&lt; processes [i].</pre></pre></pre>
	tumoround < "It" < processes [i]. waiting < "In":
	1) 5
	int main () {
	Process processes [] = {} 1,08}, {2,14}
	13,2,9}, 14,3,5}};
CT MAN	int n= 4:
31. VIIV	ICENT PALLOTTI COLLEGE OF ENGINEERING & TECHNOLOGY, NAGPUR - 441 108

Conclusion: The Round Robin scheduling Algorithm is efficient for time-showing systems a ensures fairness among processes. The selection of an aptimum time quantum is trudy for performance. Lonclusion . The Culput : Jours ! time quantum is for time shains ystems & PID Je maple round Robin | Pocesces , n int time Quantum = 5. 65 12 Round Robin whideling Algridhm Arriva 0 molesses G o۵ G

trucial for performance

Passers fairness

is efficient

et an optimum

11

T.

6

19

10 S lampletion

Waiting

80

time Muschen

5

عا \*

ST. VINCENT PALLOTTI COLLEGE OF ENGINEERING & TECHNOLOGY, NAGPUR - 441 108

(5)