# Simulador de escalonamento de processos

Trabalho Final de Sistemas Operacionais

Daniel Fernando Jimenez 117028769 Matheus Cunha Simões 117091021

## 1. Introdução

Neste trabalho vamos desenvolver um simulador em C de escalonamento de processos usando a estratégia de seleção Round Robin com Feedback, nele vamos observar o que acontece em cada instante de tempo T para poder entender melhor como o processador lida com vários processos, várias filas de entrada e de I/O.

## 2. Objetivos

- Criar um simulador em C sem uso de bibliotecas externas nem estruturas prontas, usar o basico do basico assim como o processador faz.
- Entender a estratégia de Round Robin e seu time slice para poder ser aplicada num ambiente de geração de processos aleatórios, solicitações de I/O e fila de alta e baixa prioridade

## 3. Premissas

- A quantidade de processos foi definida por 5 (Podem ser mudadas logo no começo do código).
- Os Tempos de duração de cada tipo de I/O ficaram assim (Podem ser mudadas logo no começo do código).

I/O	Tempo
Disco	5
Fita	10
Impressora	15

- O Tempo de serviço e o tempo de chegada são gerados aleatoriamente para cada processo no começo do programa.
- Cada fatia de Tempo é de 1 unidade e o time slice da estratégia ficou estabelecida em 3 (Pode ser mudada logo no começo do código).

 A solicitação de IO para cada processo sendo executada foi randômica, a cada instante de tempo ele tem um 10% de probabilidade de sair do processador e entrar numa fila IO.

## 4. Descrição

Nós dividimos o desenvolvimento do programa em vários passos, o primeiro passo foi a criação das bibliotecas que vão representar os processos, os dispositivos de IO e a fila. Para a criação da fila usamos a seguinte estrutura base

```
//A structure to represent a queue
typedef struct _Queue {
   int front, rear, size;
   unsigned capacity;
   int* array;
} Queue;
```

e nela criamos as funções de isFull, isEmpty, enqueue, dequeue, front e rear. Ela vai servir para simular nossa fila de alta e baixa prioridade para processos no estado de pronto.

No arquivo do processo fizemos algo mais simples porque nele só precisávamos da estrutura de controle

```
//A structure to represent a process
typedef struct _Process {
   int id;
   int startTime; //apenas para agendar o inicio na simulacao
   int timeExecuted;
   int executionTime;
   int endTime; //apenas para guardar o fim da simulação e calcular o turnaround
} Process;
```

Com isso conseguimos controlar o inicio do processo e o tempo de serviço no processador, adicionalmente criamos as três funções básicas para o processo que são: createProcess( criação do processo e seus atributos básicos), remaining(pra saber o tempo que falta pra sair do processador) e exec(simulando a execução).

```
Por último o IO device foi criado da seguinte maneira
```

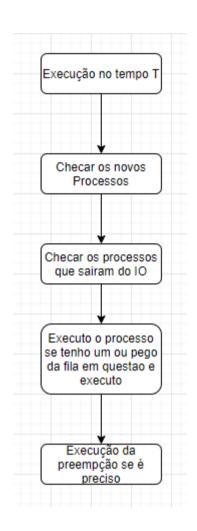
```
typedef struct _IOdevice {
   Queue *processes; //fila de processos para realizar IO
   int executed; //quanto jÃ; foi executado do processo no topo da fila
   int operationTime; //tempo da operacao de IO
}
IOdevice;
```

Com essa estrutura controlamos o processo usando o dispositivo IO e o tempo que ele está levando para executar esse IO. Além disso foram criadas 4 funções básicas de controle sobre essa estrutura que são: createlOdevice(para inicializar o dispositivo), enqueuelO(para adicionar um processo ao IO), remaininglO(Para saber quanto falta para o processo acabar) e o execlO(simular a execução do processo no IO). Adicionalmente foi criado uma variável de IOdevice para cada dispositivo de IO(Disca, Fita, Impressora)

Com todas essas estruturas básicas criadas o simulador fica mais tranquilo, porque agora é depois de inicializar todos os processos, as filas, os dispositivos IO, fazemos o seguinte fluxo:

Em cada passo da execução do tempo T:

- a. Checagem de processos novos, para ser adicionados na fila de espera.
- b. Checagem dos processos que acabaram o IO para serem adicionados na fila de espera.
- c. Caso exista um processo no processador ele vai ser executado, se não pego o primeiro processo da fila de prioridade alta, caso ela esteja vazia pego da fila de prioridade baixa e caso não tenha nada fica ocioso o processador.
- d. Se executei algum processo e o tempo de preempção é batido, retiro o processo do processador e boto ele de volta na fila.



## 5. Resultados

Por último adicionamos no simulador a geração de um arquivo csv com a tabela dos processos parecida à tabela feita nas listas desenvolvidas, segue o exemplo aberto no excel para conseguir uma formatação bonita:

С	D	Е	F	3 H	ı,	J	K	L	М	Ν	0	P	Q	R	S	Т	U	٧	W	Х	Υ	Z	AA /	AB	AC A	AD .	AE /	AF /	AG A	ΔН.	ΑI	AJ /	ΑK	AL	AM.	AN	40	AP .	AQ /	٩R	AS	ΑТ	ΑU	ΑV	ΑW	AX	A'
2	3	4	5 (	5 7	7 8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	5
							1	1	1	1	1	1	1	1	1				1	1	1																										
																																					2	2	2	Δ.	Α	Α	Α	Α			
																																		3	3	3										3	
																															4	4	4										4	4	4		
																5	5	5				5	5	5	5	5	5	Α ,	A A	Δ /	Δ ,	Α								5	5	5					
BA	A E	ВВ	В	CE	BD	BE	BF	В	GE	ВН	ВІ	BJ	BK	BL	BN	ИΒ	NB	30	ВР	BQ	BR	BS	ВТ	BU	J B	V B	WE	3X	BY	ΒZ	CA	CI	ВС	C	D	CE	CF	CG	СН	C	ı	IJ (	CK	CL	CN	CN	1
52	2 .	53	54	4 !	55	56	57	7 5	8 5	59	60	61	62	63	64	1 6	5 6	66	67	68	69	70	71	7.	2 7	3 7	74	75	76	77	78	79	9 8	80 8	31	82	83	84	85	8	6 8	7	88	89	90	91	ı
:	2	2	:	2							2	2	2		2	2	2	2		2	2	2	2	2													С	С	С	С	С	(	С	С	С	С	(
									3	3	Α	Α	Α	Α	Α				3	С	С	С	С	С	С	C	. (	3	С	С	С	С	C	C	: (	С	3	3	3					3	3	3	3
					4			4						4	п	п			n	D	D	n	D	D	D		4	4	4		-		4	4	4	4					4	4	4				
	B,	BA 52	BA BB 52 53	BA BB BC 52 53 54	BA BB BC E 52 53 54 \$	BA BB BC BD 52 53 54 55	BA BB BC BD BE 52 53 54 55 56	BA BB BC BD BE BI 52 53 54 55 56 5	BA BB BC BD BE BF B 52 53 54 55 56 57 5	BA BB BC BD BE BF BG E 52 53 54 55 56 57 58 \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$	BA BB BC BD BE BF BG BH 52 53 54 55 56 57 58 59 2 2 2 3 3 3 3	BA BB BC BD BE BF BG BH BI 52 53 54 55 56 57 58 59 60 2 2 2 2	BA BB BC BD BE BF BG BH BI BJ 52 53 54 55 56 57 58 59 60 61 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	BA BB BC BD BE BF BG BH BI BJ BK 52 53 54 55 56 57 58 59 60 61 62 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	2       3       4       5       6       7       8       9       10       11       12       13       14       15       16       17         BA       BB       BC       BD       BE       BF       BG       BH       BI       BJ       BK       BL         52       53       54       55       56       57       58       59       60       61       62       63         2       2       2       2       2       2       2       2         3       3       A       A       A       A       A       A	BA BB BC BD BE BF BG BH BI BJ BK BL BN 52 53 54 55 56 57 58 59 60 61 62 63 64 64 64 64 64 64 64 64 64 64 64 64 64	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 1 1 1 1 1 1 1 1 1 1 1 1 5  BA BB BC BD BE BF BG BH BI BJ BK BL BM BI 52 53 54 55 56 57 58 59 60 61 62 63 64 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 1 1 1 1 1 1 1 1 1 1 1 1 5 5  BA BB BC BD BE BF BG BH BI BJ BK BL BM BN E 52 53 54 55 56 57 58 59 60 61 62 63 64 65 6  2 2 2 2 2 2 2 2 2 2 2 3 3 3 A A A A A A	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 5 5 5 5  BA BB BC BD BE BF BG BH BI BJ BK BL BM BN BO 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30  BA BB BC BD BE BF BG BH BI BJ BK BL BM BN BO BP BQ BR BS BT BU BV BW B 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 22 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31  BA BB BC BD BE BF BG BH BI BJ BK BL BMBN BO BP BQ BR BS BT BU BV BW BX  5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32  BA BB BC BD BE BF BG BH BI BJ BK BL BMBN BO BP BQ BR BS BT BU BV BW BX BY  5 5 5 5 6 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76  2 2 2 2 3 3 3 3 A A A A A A A A A A A A	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38  BA BB BC BD BE BF BG BH BI BJ BK BL BM BN BO BP BQ BR BS BT BU BV BW BX BY BZ CA CB CC CD  52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81  2 2 2 2 3 2 6 2 6 2 7 2 8 2 9 30 31 32 33 34 35 36 37 38	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39  BA BB BC BD BE BF BG BH BI BJ BK BL BM BN BO BP BQ BR BS BT BU BV BW BX BY BZ CA CB CC CD CE  52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82  2 2 2 2 3 3 3 4 A A A A A A A A A A A A A A A A	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40  2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40  2 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40  2 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40  2 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40  2 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40  2 2 3 4 5 6 7 8 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40  2 2 3 4 5 6 7 8 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40  3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41  BA BB BC BD BE BF BG BH BI BJ BK BL BM BN BO BP BQ BR BS BT BU BV BW BX BY BZ CA CB CC CD CE CF CG 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42    A	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43    A	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44    1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 46 46 46 46 46 46 46 46 46 46 46 46	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47    A   A   A   A   A   A   A   A   A	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48  BA BB BC BD BE BF BG BH BI BJ BK BL BM BN BO BP BQ BR BS BT BU BV BW BX BY BZ CA CB CC CD CE CF CG CH CI CJ CK CL CM  52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

#### Criando processos:

- Processo 1: Chegada no tempo = 10, tempo de servico = 12.
- Processo 2: Chegada no tempo = 38, tempo de servico = 23.
- Processo 3: Chegada no tempo = 37, tempo de servico = 17.
- Processo 4: Chegada no tempo = 34, tempo de servico = 22.
- Processo 5: Chegada no tempo = 19, tempo de servico = 12.
- t = 0: CPU ociosa.
- t = 1: CPU ociosa.
- t = 2: CPU ociosa.
- t = 3: CPU ociosa.
- t = 4: CPU ociosa.
- t = 5: CPU ociosa.
- t = 6: CPU ociosa.
- t = 7: CPU ociosa.
- t = 8: CPU ociosa.
- t = 9: CPU ociosa.
- t = 10: Processo 1 foi iniciado e inserido na fila de alta prioridade.
- t = 10: Processo 1 retirado da fila de alta prioridade e entra em execucao.
- t = 13: Processo 1 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 13: Processo 1 retirado da fila de baixa prioridade e entra em execucao.
- t = 16: Processo 1 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 16: Processo 1 retirado da fila de baixa prioridade e entra em execucao.
- t = 19: Processo 1 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 19: Processo 5 foi iniciado e inserido na fila de alta prioridade.
- t = 19: Processo 5 retirado da fila de alta prioridade e entra em execucao.
- t = 22: Processo 5 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 22: Processo 1 retirado da fila de baixa prioridade e entra em execucao.
- t = 25: O processo 1 finalizou sua execucao.
- t = 25: Processo 5 retirado da fila de baixa prioridade e entra em execucao.
- t = 28: Processo 5 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 28: Processo 5 retirado da fila de baixa prioridade e entra em execução.
- t = 31: Processo 5 inserido na fila de I/O para disco.
- t = 31: CPU ociosa.
- t = 32: CPU ociosa.
- t = 33: CPU ociosa.
- t = 34: Processo 4 foi iniciado e inserido na fila de alta prioridade.
- t = 34: Processo 4 retirado da fila de alta prioridade e entra em execucao.
- t = 36: Processo 5 terminou I/O em disco e foi inserido na fila de baixa prioridade.
- t = 37: Processo 4 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 37: Processo 3 foi iniciado e inserido na fila de alta prioridade.
- t = 37: Processo 3 retirado da fila de alta prioridade e entra em execucao.
- t = 38: Processo 2 foi iniciado e inserido na fila de alta prioridade.
- t = 40: Processo 3 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 40: Processo 2 retirado da fila de alta prioridade e entra em execucao.
- t = 43: Processo 2 inserido na fila de I/O para disco.
- t = 43: Processo 5 retirado da fila de baixa prioridade e entra em execucao.
- t = 46: O processo 5 finalizou sua execucao.

- t = 46: Processo 4 retirado da fila de baixa prioridade e entra em execucao.
- t = 48: Processo 2 terminou I/O em disco e foi inserido na fila de baixa prioridade.
- t = 49: Processo 4 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 49: Processo 3 retirado da fila de baixa prioridade e entra em execucao.
- t = 52: Processo 3 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 52: Processo 2 retirado da fila de baixa prioridade e entra em execução.
- t = 55: Processo 2 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 55: Processo 4 retirado da fila de baixa prioridade e entra em execucao.
- t = 58: Processo 4 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 58: Processo 3 retirado da fila de baixa prioridade e entra em execucao.
- t = 60: Processo 3 inserido na fila de I/O para disco.
- t = 60: Processo 2 retirado da fila de baixa prioridade e entra em execucao.
- t = 63: Processo 2 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 63: Processo 4 retirado da fila de baixa prioridade e entra em execucao.
- t = 64: Processo 4 inserido na fila de I/O para fita magnetica.
- t = 64: Processo 2 retirado da fila de baixa prioridade e entra em execucao.
- t = 65: Processo 3 terminou I/O em disco e foi inserido na fila de baixa prioridade.
- t = 67: Processo 2 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 67: Processo 3 retirado da fila de baixa prioridade e entra em execucao.
- t = 68: Processo 3 inserido na fila de I/O para impressora.
- t = 68: Processo 2 retirado da fila de baixa prioridade e entra em execução.
- t = 71: Processo 2 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 71: Processo 2 retirado da fila de baixa prioridade e entra em execucao.
- t = 72: Processo 2 inserido na fila de I/O para impressora.
- t = 72: CPU ociosa.
- t = 73: CPU ociosa.
- t = 74: Processo 4 terminou I/O em fita magnetica e foi inserido na fila de alta prioridade.
- t = 74: Processo 4 retirado da fila de alta prioridade e entra em execucao.
- t = 77: Processo 4 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 77: Processo 4 retirado da fila de baixa prioridade e entra em execucao.
- t = 80: Processo 4 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 80: Processo 4 retirado da fila de baixa prioridade e entra em execucao.
- t = 83: Processo 4 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 83: Processo 3 terminou I/O em impressora e foi inserido na fila de alta prioridade.
- t = 83: Processo 3 retirado da fila de alta prioridade e entra em execucao.
- t = 86: Processo 3 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 86: Processo 4 retirado da fila de baixa prioridade e entra em execucao.
- t = 89: O processo 4 finalizou sua execucao.
- t = 89: Processo 3 retirado da fila de baixa prioridade e entra em execucao.
- t = 92: Processo 3 sofreu preempcao, inserido na fila de baixa prioridade.
- t = 92: Processo 3 retirado da fila de baixa prioridade e entra em execucao.
- t = 94: O processo 3 finalizou sua execucao.
- t = 94: CPU ociosa.
- t = 95: CPU ociosa.
- t = 96: CPU ociosa.
- t = 97: CPU ociosa.
- t = 98: Processo 2 terminou I/O em impressora e foi inserido na fila de alta prioridade.
- t = 98: Processo 2 retirado da fila de alta prioridade e entra em execucao.

t = 101: Processo 2 sofreu preempcao, inserido na fila de baixa prioridade.

t = 101: Processo 2 retirado da fila de baixa prioridade e entra em execucao.

t = 104: Processo 2 sofreu preempcao, inserido na fila de baixa prioridade.

t = 104: Processo 2 retirado da fila de baixa prioridade e entra em execucao.

t = 105: O processo 2 finalizou sua execucao.

## Р1

Chegada:10

Tempo de Servico:12 Tempo de espera:13 Turnaround(Tq):15

Desempenho(Tq/Ts):1.250000

## P2

Chegada:38

Tempo de Servico:23 Tempo de espera:82 Turnaround(Tq):67 Desempenho(Tq/Ts):2.913043

## P3

Chegada:37

Tempo de Servico:17 Tempo de espera:77 Turnaround(Tq):57

Desempenho(Tq/Ts):3.352941

## P4

Chegada:34

Tempo de Servico:22 Tempo de espera:67 Turnaround(Tq):55

Desempenho(Tq/Ts):2.500000

#### P5

Chegada:19

Tempo de Servico:12 Tempo de espera:34 Turnaround(Tq):27

Desempenho(Tq/Ts):2.250000

# 6. Conclusões

 Nas diferentes simulações feitas conseguimos entender o conceito de starvation na hora que um processo na fila baixa esperando ser atendido pelo processador, tem que ficar esperando pois chegam novos processos na fila de baixa prioridade.

•	O estratégia Round Robin atende todos os processos de maneira justa, permitindo que processos curtos sejam processados sem esperar por outros processos longos que chegaram antes que ele.