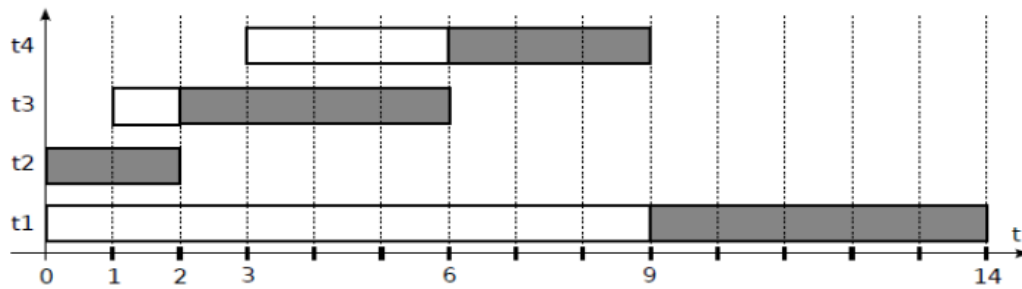


O algoritmo de escalonamento que proporciona os **menores tempos médios de execução e de espera** é conhecido como menor tarefa primeiro, ou SJF (Shortest Job First). Consiste em atribuir o processador à menor (mais curta) tarefa da fila de tarefas prontas.



<http://nasemanadaprova.blogspot.com/2015/12/algoritmos-de-escalonamento-de-processos.html>

Nesse momento de decidir qual escalonador será utilizado no sistema operacional, cabe avaliar o cenário que o sistema será utilizado.

Deve-se ter cuidado com algumas variáveis como em casos que necessitam de mais processamento, ou seja, ação da CPU. Como com processos que necessitam de processamento, ocuparão a CPU por um tempo maior e não precisarão, ou de pouca, intervenção do usuário.

Enquanto isso, há processos que necessitam de mais entrada e saída de dados, ou seja, o processo necessita de intervenção do usuário.

Ou seja, deve-se verificar o comportamento dos processos: se são orientados a Entrada e Saída (IN/OUT bound) ou orientados a orientados a CPU (CPU bound).

<https://www.oficinadanet.com.br/post/12781-sistemas-operacionais-o-que-e-escalonamento-de-processos>

O escalonador ideal é aquele que consegue deixar a CPU 100% ocupada para maximizar a produtividade e minimizar o tempo de retorno, resposta e espera.

Não existe nenhuma política de escalonamento ótima. A política de escalonamento conveniente depende do tipo de processo e do critério de otimização desejado

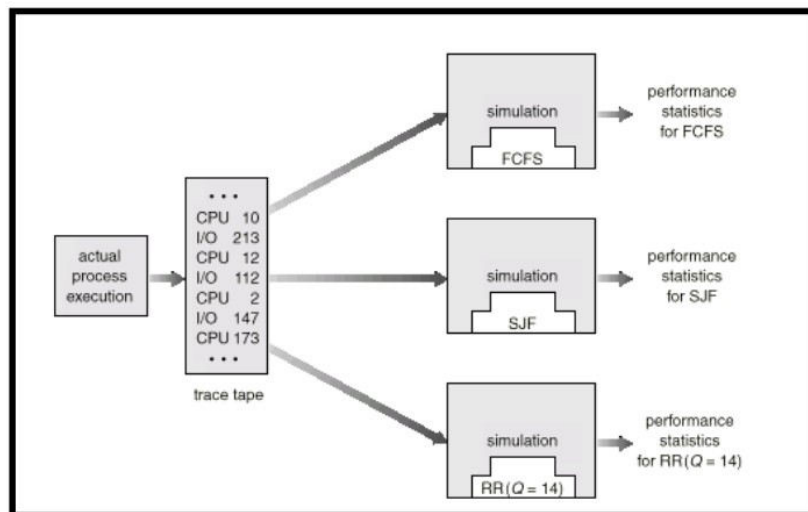
Maneiras de se fazer avaliação dos algoritmos de escalonamento:

**Modelagem determinística:** considera uma carga de trabalho particular (pré-determinada) Define (calcula) o desempenho de cada algoritmo para a carga utilização de CPU, throughput, tempo de espera, tempo de turnaround, etc

**Avaliação por simulação:** método mais preciso, utiliza um modelo de sistema de computação, as informações (processos, picos de CPU, chegadas, E/S, termos, etc.) podem ser geradas aleatoriamente. Resultados são usados para verificar o que ocorre na realidade e adota-se a distribuição adequada

**Avaliação por implementação:** mais realista porém tem alto custo: necessário implementar no kernel e testar sob as diversas situações reais

Exemplo de avaliação por simulação:



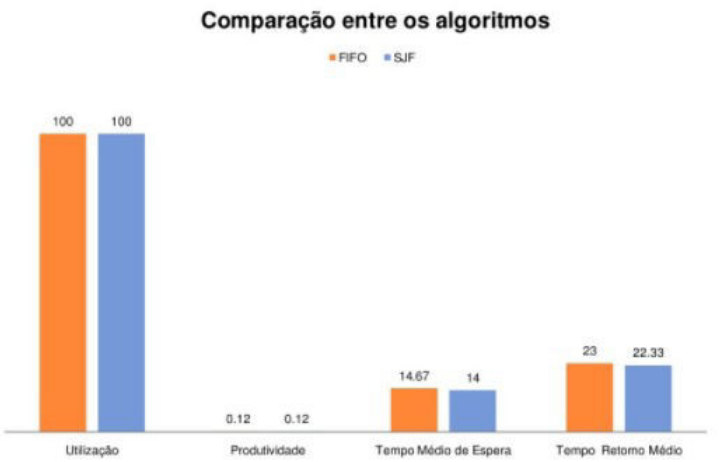
<http://www.univasf.edu.br/~andreza.leite/aulas/SO/ProcessosEscalonamento.pdf>

Pode-se observar que os **algoritmos preemptivos** (RR, SRTF e PRIOp) possuem um número de trocas de contexto maior que seus correspondentes **cooperativos (ou não-preemptivos)**, o que era de se esperar. Também pode-se constatar que o algoritmo SRTF proporciona os melhores tempos médios de execução  $T_t$  e de espera  $T_w$ , enquanto os piores tempos são providos pelo algoritmo RR (que, no entanto, oferece um melhor tempo de resposta a aplicações interativas). Observa-se também que o tempo total de processamento é constante, pois ele só depende da carga de processamento de cada tarefa e não da ordem em que são executadas. Contudo, esse tempo pode ser influenciado pelo número de trocas de contexto, caso seja muito elevado.

Algoritmo de escalonamento	FCFS	RR	SJF	SRTF	PRIOc	PRIOp	PRIOd
Tempo médio de execução $T_t$	8,0	8,4	5,8	5,4	6,6	5,6	5,8
Tempo médio de espera $T_w$	5,2	5,6	3,0	2,6	3,8	2,8	3,0
Número de trocas de contexto	4	7	4	5	4	6	6
Tempo total de processamento	14	14	14	14	14	14	14

<http://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=socm:socm-06.pdf>

# COMPARAÇÃO ENTRE OS ALGORITMOS

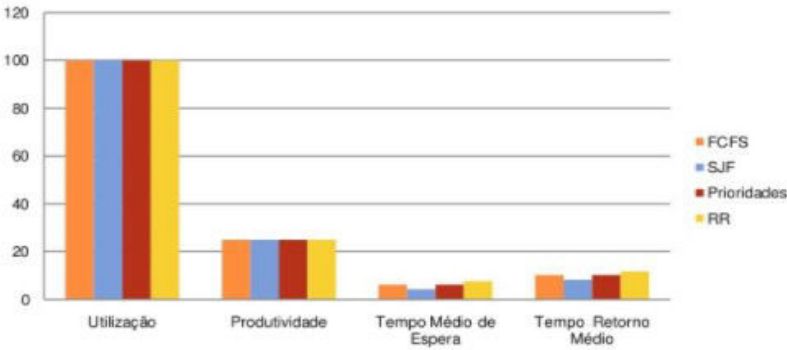


## CENÁRIO 1

Processo	Instante de Chegada	Tempo de Execução	Prioridade
P1	0	3	1
P2	0	5	1
P3	0	6	1
P4	0	2	1

Algoritmo	FIFO	SJF	Prioridades	RR
Utilização da CPU(%)	100	100	100	100
Produtividade da CPU(%)	25	25	25	25
Tempo Médio de Espera	6,25	4,25	6,25	7,75
Tempo Médio de Retorno	10,25	8,25	10,25	11,75

Cenário 1



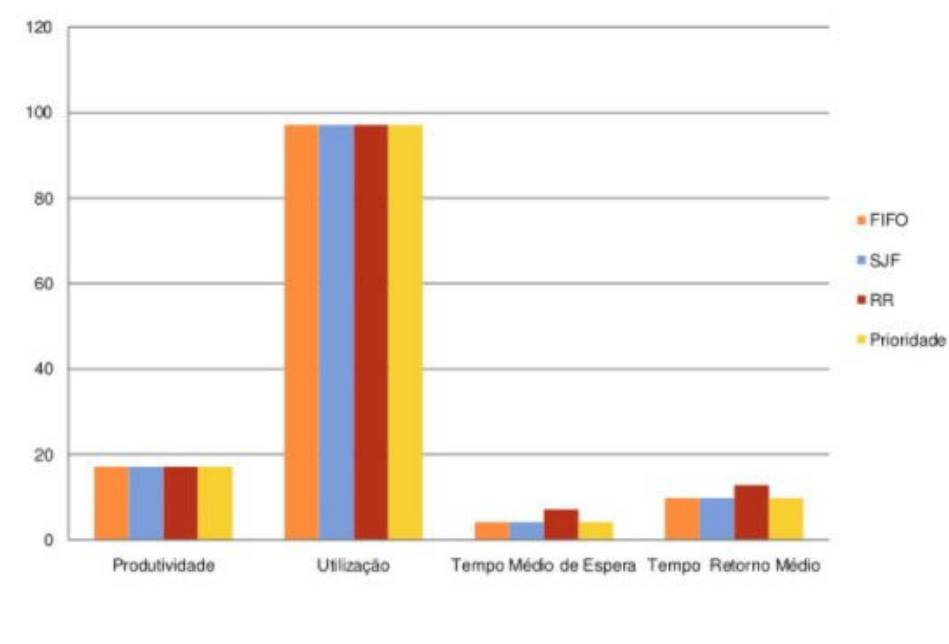
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FC-FS	1	1	1	2	2	2	2	2	3	3	3	3	3	3	4	4
SJF	4	4	1	1	1	2	2	2	2	2	3	3	3	3	3	3
FPP	1	1	1	2	2	2	2	2	3	3	3	3	3	3	4	4
RR	1	2	3	4	1	2	3	4	1	2	3	2	3	2	3	3

CENÁRIO 2

Processo	Instante de Chegada	Tempo de Execução	Prioridade
P1	5	6	2
P2	3	3	2
P3	1	2	2
P4	2	1	2
P5	7	10	2
P6	8	12	2

Algoritmo	FCFS	SJF	Prioridades	RR
Utilização da CPU(%)	97,14	97,14	97,14	97,14
Produtividade da CPU(%)	17,14	17,14	17,14	17,14
Tempo Médio de Espera	4,17	4,17	7,17	12
Tempo Médio de Retorno	9,83	9,83	17,67	12,83

Cenário 2

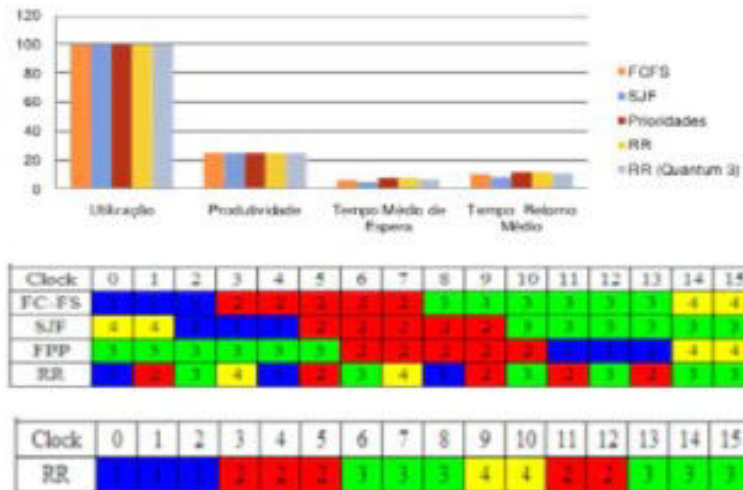


CENÁRIO 3

Processo	Instante de Chegada	Tempo de Execução	Prioridade
P1	0	3	3
P2	0	5	2
P3	0	6	1
P4	0	2	4

Algoritmos	FCFS	SJF	Prioridades	RR	RR( Q 3)
Utilização da CPU(%)	100	100	100	100	100
Produtividade da CPU(%)	25	25	25	25	25
Tempo Médio de Espera	6,25	4,25	7,75	7,75	6,75
Tempo Médio de Retorno	10,25	8,25	11,75	11,75	10,75

### Cenário 3

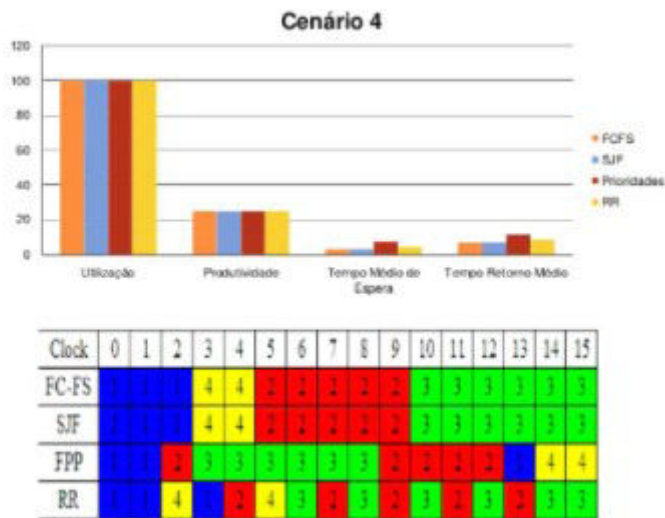


### CENÁRIO 4

Processo	Instante de Chegada	Tempo de Execução	Prioridade
P1	0	3	3
P2	2	5	2
P3	3	6	1
P4	1	2	4

Algoritmo	FCFS	SJF	Prioridades	RR
Utilização da CPU(%)	100	100	100	100
Produtividade da CPU(%)	25	25	25	25
Tempo Médio de Espera	3	3	7,5	4,5
Tempo Médio de Retorno	7	7	11,5	8,5



Essas simulações foram feitas por um grupo de alunos por meio de dois simuladores:

->SOSIM

→ SimulaRSO

<https://pt.slideshare.net/TallesNascimentoRodrigues/sistemas-operacionais-escalonamento-de-processos>

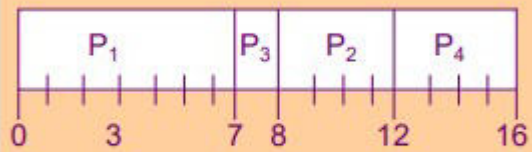
Indicador	Classificação														
Tempo de Espera	SJF	«	SRT	«	IFCS	«	PFCS	«	FCFS	«	FPCS	«	PRTY	«	RR
Throughput	SJF	«	SRT	«	PFCS	«	IFCS	«	FPCS	«	PRTY	«	FCFS	«	RR
Turnaround	SRT	«	SJF	«	IFCS	«	PFCS	«	PRTY	«	FPCS	«	FCFS	«	RR
Utilização CPU	RR	«	SJF	«	FPCS	«	FCFS	«	PRTY	«	PFCS	«	SRT	«	IFCS

[https://www.formiga.ifmg.edu.br/documents/2018/Biblioteca/TCCs\\_e\\_Artigos/Danilo\\_da\\_Silva\\_Alves.pdf](https://www.formiga.ifmg.edu.br/documents/2018/Biblioteca/TCCs_e_Artigos/Danilo_da_Silva_Alves.pdf)

### SJF não-preemptivo

Processo	Arrival Time	Burst Time
$P_1$	0.0	7
$P_2$	2.0	4
$P_3$	4.0	1
$P_4$	5.0	4

□ SJF (não-preemptivo)



□ Tempo médio de espera  
 $(0 + 6 + 3 + 7)/4 = 4$

### SJF preemptivo

Processo	Arrival Time	Burst Time
$P_1$	0.0	7
$P_2$	2.0	4
$P_3$	4.0	1
$P_4$	5.0	4

□ SJF (preemptivo)



□ Tempo médio de espera  
 $(9 + 1 + 0 + 2)/4 = 3$