



Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Disciplina: Sistemas Operacionais I

Aula 15: Escalonamento de CPU P1

Prof. Diogo Branquinho Ramos

diogo.branquinho@fatec.sp.gov.br

São José dos Campos - SP

Roteiro

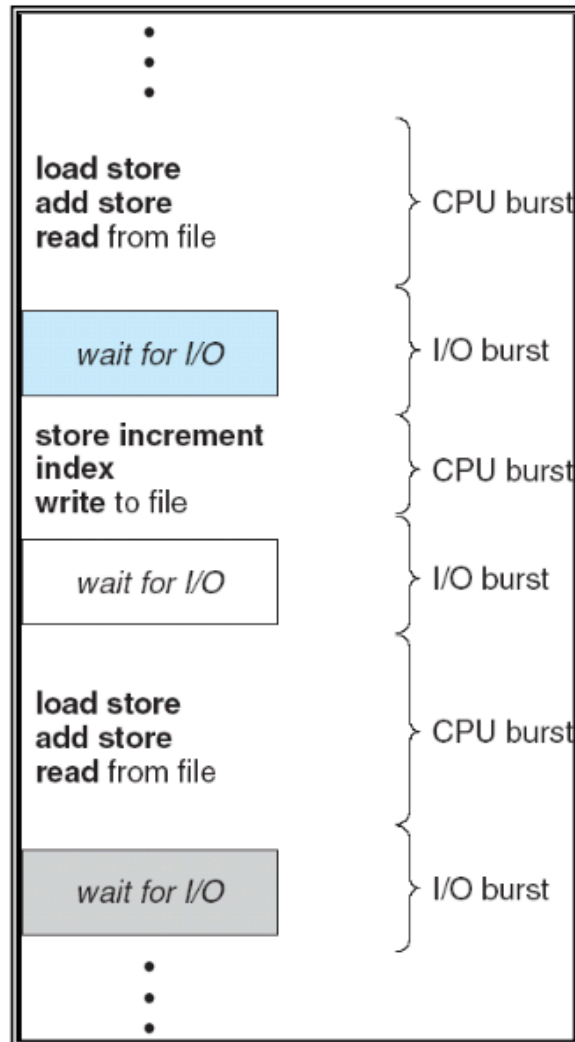
- Conceitos básicos
- Critérios de escalonamento
- Algoritmos de escalonamento

Conceitos básicos

- **Necessidade de escalonar: multiprogramação!**
 - Utilização máxima de CPU obtida com multiprogramação.
- **Execução de um processo**
 - Um ciclo de execução de CPU e espera de E/S.
 - Começa com um burst de CPU (uso de CPU), que é seguido por um burst de E/S (uso de E/S), que é seguido por um burst de CPU...

Sequência alternada de bursts de CPU e E/S

Execução típica



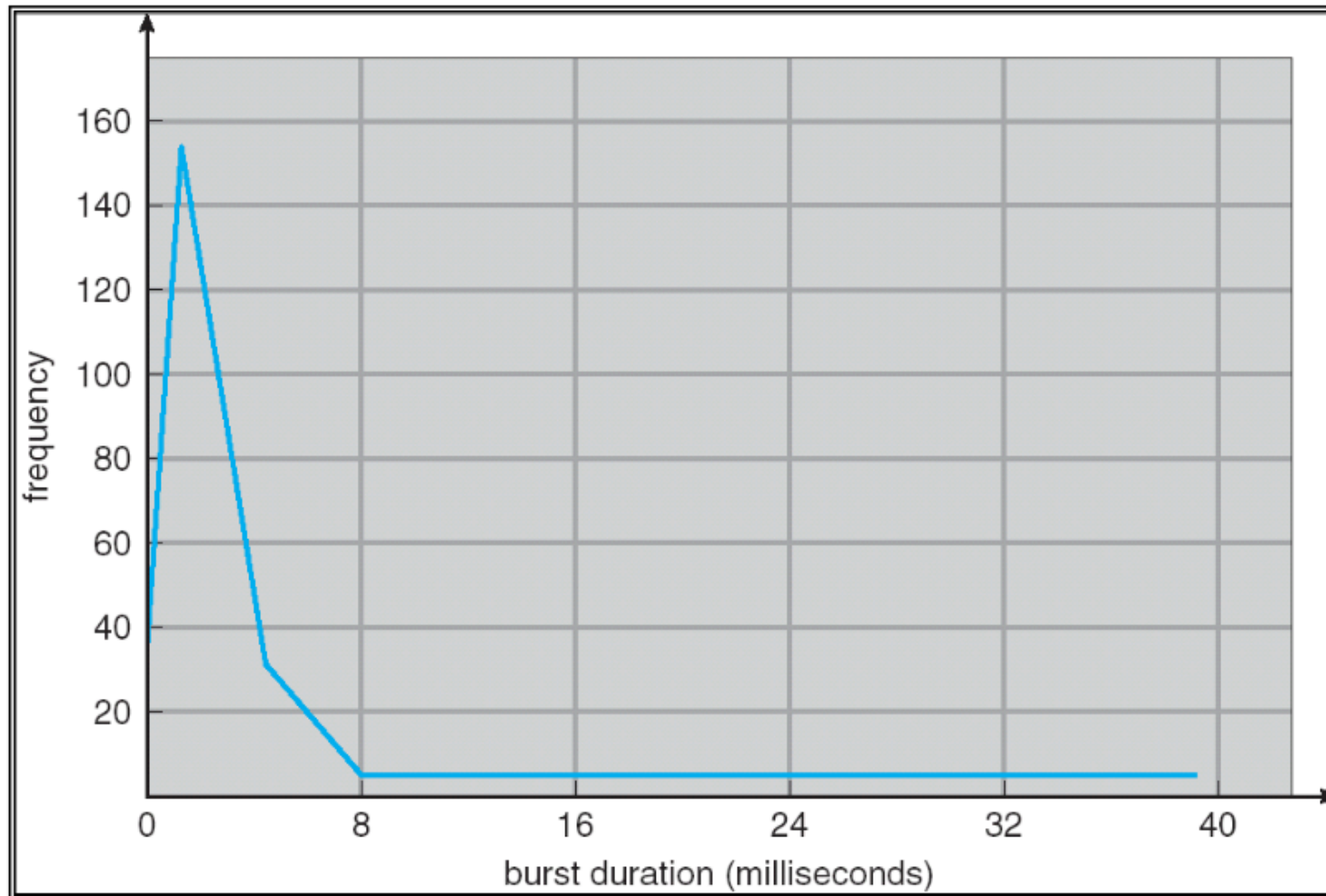
Histograma de tempos de burst de CPU

Sistema genérico

- **Curva de frequência**
 - Exponencial
 - Grande quantidade de bursts de CPU curtos;
 - Pequena quantidade de bursts de CPU longos.
 - I/O-bound
 - Bursts de CPU curtos (muitos).
 - CPU-bound
 - Bursts de CPU longos (alguns).
- Essa informação pode ser importante na seleção de um algoritmo de escalonamento!

Histograma de tempos de burst de CPU

Sistema genérico



Escalonador de CPU

- **Quem executa?**
 - Escalonador de curto prazo.
- **Funcionamento**
 - Seleciona dentre os processos na memória que estejam prontos para executar, e aloca a CPU a um deles.
- **A fila de prontos não é necessariamente uma fila FIFO**
 - Depende do algoritmo de escalonamento.
 - Os registros na fila são os PCBs.

Preempção

- **Definição**

- Escalonamento preemptivo: o processo pode perder direito à CPU subitamente – exige hardware temporizador.
- Escalonamento não preemptivo: o processo libera a CPU quando termina ou se precisar de E/S.

- **Algumas preocupações**

- Dados compartilhados entre dois processos.
 - Podem ficar em estado incoerente. Solução: sincronismo.
- Syscall preemptada: mudanças de dados do kernel.
 - SOLUÇÃO: syscall atômica. Kernel não preempta com dados em estado incoerente.

Cr terios de escalonamento

- **Cada algoritmo de escalonamento possui suas propriedades peculiares.**
 - A escolha de um algoritmo pode favorecer uma classe de processos em detrimento de outra.
- **  preciso considerar cr terios para escolher melhor algoritmo!**

Critérios de escalonamento

- **Utilização de CPU**
 - Manter a CPU a maior parte do tempo ocupada.
- **Throughput (vazão)**
 - # de processos que completam sua execução por unidade de tempo.
- **Tempo de turnaround (tempo de retorno)**
 - Quantidade de tempo para executar um processo em particular, desde o carregamento na memória até o término da execução.
 - Limitado por E/S.

Critérios de escalonamento

- **Tempo de espera**
 - Todo o tempo em que um processo esteve esperando na fila de prontos
 - Não computa os tempos de espera por E/S.
- **Tempo de resposta**
 - Tempo desde quando uma solicitação foi submetida até a primeira resposta ser produzida. Tempo gasto para começar a responder.
 - Pode depender da chegada de dados na CPU vindos da cache ou RAM.

Critérios de escalonamento

- **Equação**
 - **Maximizar** a utilização e o throughput da CPU e
 - **Reduzir** o turnaround, o tempo de espera e o tempo de resposta.

Algoritmos de escalonamento

First-Come, First-Served (FCFS)

- **Fácil de implementar**
- **Implementação da política por meio de uma FIFO**
 - O PCB do processo que entra na fila de prontos é associado ao final da fila.
 - A CPU é alocada ao processo cujo PCB está na cabeça da fila.
- **Não-preemptivo**
 - Processo mantém a CPU até o final.
- **O tempo médio de espera geralmente não é mínimo**
 - Depende dos tempos de bursts dos processos.

Algoritmos de escalonamento

First-Come, First-Served (FCFS)

Processo	Tempo de burst
----------	----------------

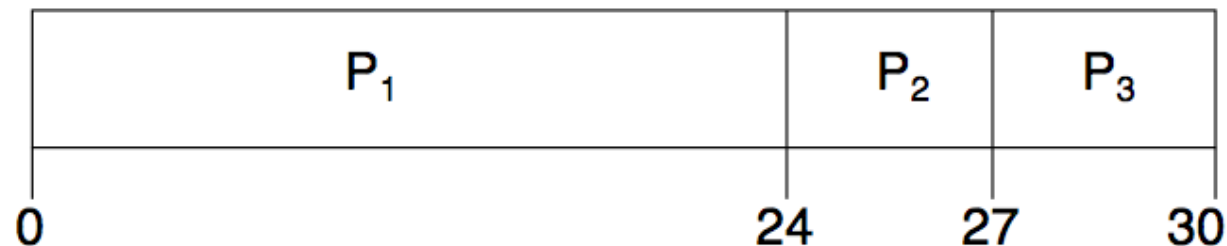
P1	24
----	----

P2	3
----	---

P3	3
----	---

Suponha que os processos cheguem nesta ordem: P1 , P2 , P3.

Diagrama de Gantt:



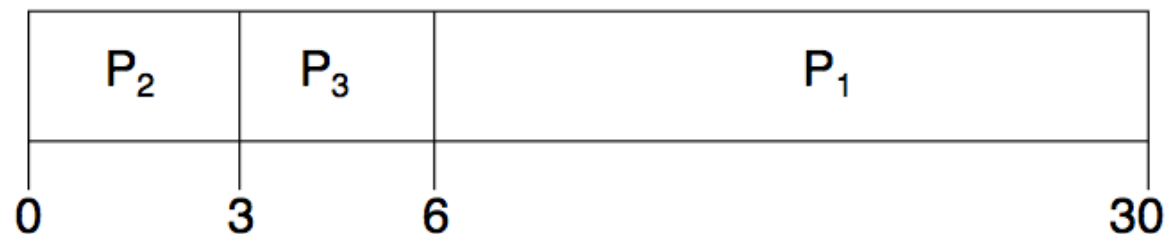
Tempo de espera para P1 = 0; P2 = 24; P3 = 27

Tempo de espera médio: $(0 + 24 + 27)/3 = 17$

Algoritmos de escalonamento

First-Come, First-Served (FCFS)

- Suponha que os processos cheguem nesta ordem: P2 , P3 , P1
- O diagrama de Gantt para o escalonamento é:



- Tempo de espera para P1 = 6; P2 = 0; P3 = 3
- Tempo de espera médio: $(6 + 0 + 3)/3 = 3$
- Muito melhor que o caso anterior!
 - Varia muito para bursts variados.
- **Efeito comboio: processo curto atrás de processo longo.**
 - Um processo CPU-bound, vários processos I/O bound.
 - Solução: priorizar processos I/O bound no uso de CPU.

Algoritmos de escalonamento

Shortest Job First (SJF)

- **Funcionamento**

- Associa a cada processo o tamanho de seu próximo burst de CPU, para escalonar o processo com o menor tempo.

Processo	Tempo de burst
----------	----------------

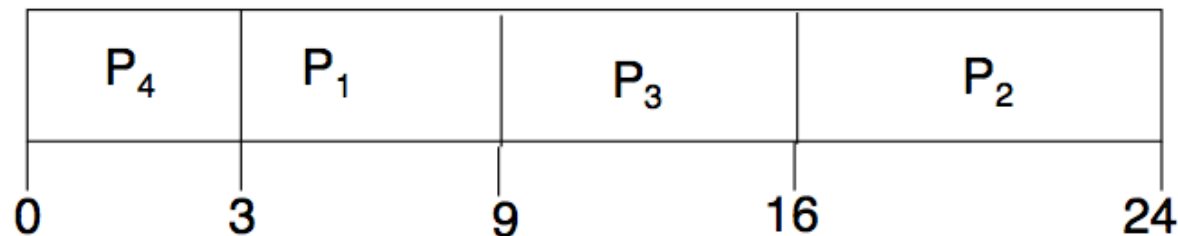
P1	6
----	---

P2	8
----	---

P3	7
----	---

P4	3
----	---

- **Como ficaria?**



Algoritmos de escalonamento

Shortest Job First (SJF)

- Tempo de espera médio SJF: 7,00
- Tempo de espera médio FCFS: 10,25
- SJF é melhor!
 - Provê o menor tempo de espera médio.
 - Estratégia: mover um processo curto antes de um longo diminui mais o tempo de espera do processo curto do que aumenta o tempo de espera de um processo longo.
 - Equação: $\frac{3a + 2b + c}{4}$

Modificação no algoritmo SJF: preempção

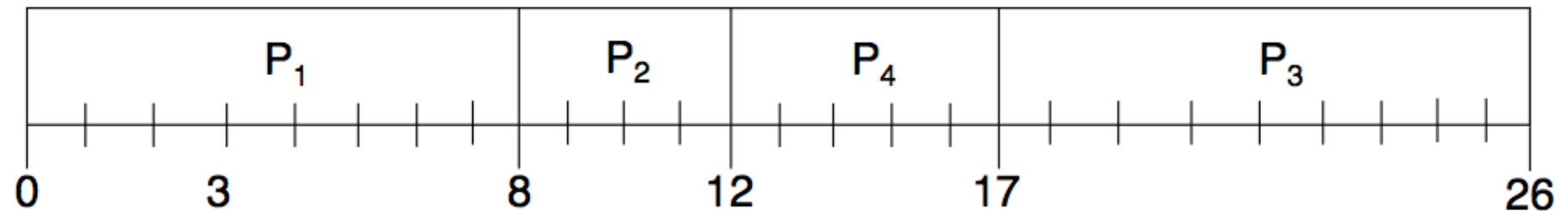
Shortest-Remaining-Time-First (SRTF)

- **Novo esquema**
 - Preemptivo: se um novo processo chega com tamanho de burst de CPU menor que o tempo restante do processo atualmente em execução, a CPU é alocada.
- **Até agora os processos chegam juntos.**
 - E se eles chegassem em tempos diferentes?

Exemplo de SJF

Processo	Tempo chegada	Tempo de burst
P1	0	8
P2	1	4
P3	2	9
P4	3	5

- SJF

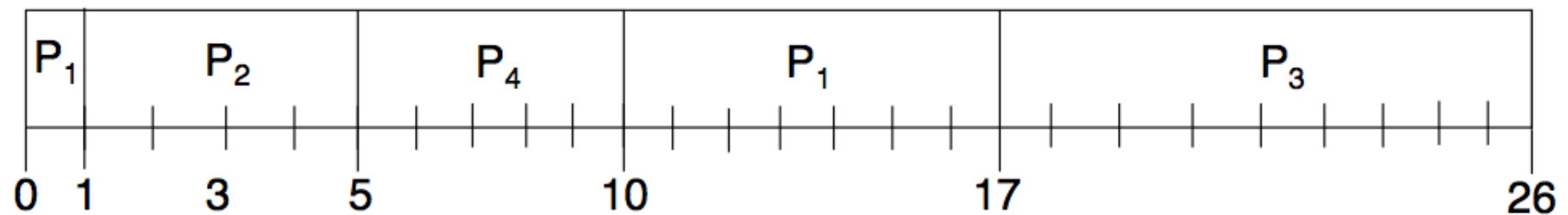


- Tempo médio de espera = $(0 + 7 + 9 + 15)/4 = 7,75$

Exemplo de SRTF

Processo	Tempo chegada	Tempo de burst
P1	0	8
P2	1	4
P3	2	9
P4	3	5

- SRTF

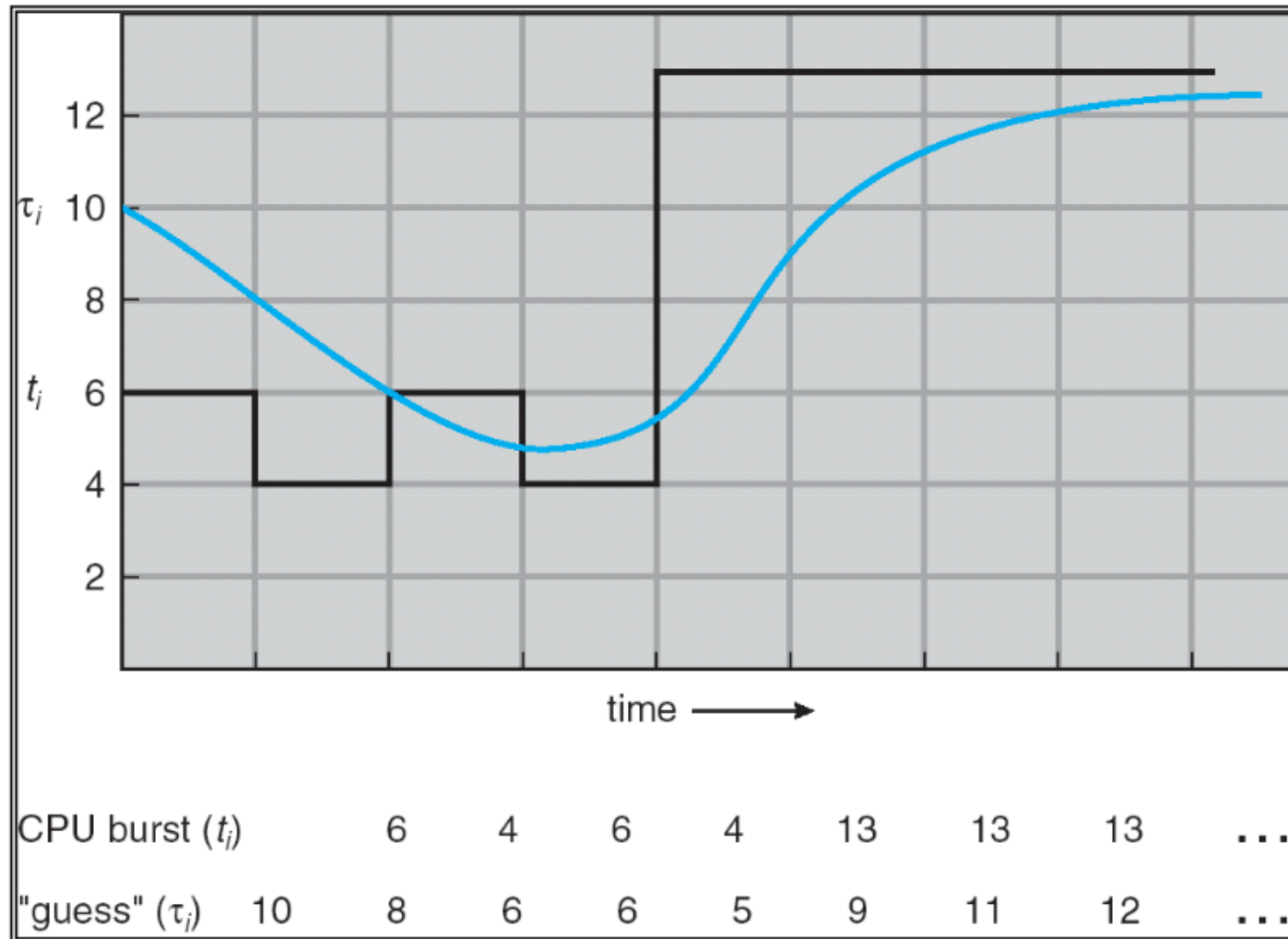


- Tempo médio de espera = $((0 + 9) + 0 + 15 + 2)/4 = 6,5$

Tamanho do próximo burst de CPU

- **Como saber o tamanho do burst de CPU antes de executar?**
- Através de estimativa usando o tamanho dos bursts de CPU anteriores e a média exponencial.
 1. t_n = extensão real do n^{th} burst de CPU
 2. τ_{n+1} = valor previsto para o próximo burst de CPU
 3. $\alpha, 0 \leq \alpha \leq 1$
 4. Define: $\tau_{n+1} = \alpha t_n + (1 - \alpha)\tau_n$.

Previsão do próximo burst de CPU



Escalonamento por prioridade

- Um número inteiro é associado a cada processo.
- A CPU é alocada ao processo com a maior prioridade (menor inteiro \equiv maior prioridade).
 - Preemptivo
 - Não preemptivo
- **SJF é um escalonamento por prioridade**
 - Será o desempate entre processos com mesma prioridade.
- **Prioridades**
 - Internamente: uma quantidade mensurável (limites de tempo, memória, número de arquivos abertos...).
 - Externamente: fora do SO (importância do processo, quantidade paga pelo uso da CPU, fatores políticos...).

Escalonamento por prioridade

- **Problema: Estagnação (*Starvation*)**
 - Processos com baixa prioridade podem nunca ser executados.
 - O IBM 7094 no MIT foi desativado em 1973 e tinha um processo de baixa prioridade submetido em 1967 ainda não executado!
- **Solução: Envelhecimento (*Aging*)**
 - À medida que o tempo passa, aumenta a prioridade do processo.
 - Um processo de prioridade 127 (baixa) chega à prioridade 0 (alta) em quase 32 horas, se sua prioridade aumentar em 1 a cada 15 min.