

Relatório

Algoritmo Round Robin com Feedback

Bruna Pimenta Ribeiro

118171816

Eduardo Freitas de Carvalho

120047675

Lucas Fernandes Silva

120135364

18 de novembro de 2024



Sumário

1. [Introdução](#)
2. [Escopo do Projeto](#)
3. [Metodologia de Implementação](#)
4. [Implementação](#)
5. [Resultados](#)
6. [Discussões dos Resultados](#)
7. [Conclusão](#)
8. [Referências](#)
9. [Apêndice](#)



Introdução

O objetivo deste trabalho é implementar um simulador para escalonamento de processos utilizando o algoritmo Round Robin com Feedback. O trabalho vai explorar **conceitos de escalonamento** em sistemas operacionais, simulando como os processos são gerenciados em filas de **prioridades diferentes** e como interagem com dispositivos de entrada/saída. Será observado na prática como o escalonador organiza e distribui os recursos entre os processos de forma **justa e eficiente**.

Escopo do Projeto

Características

De acordo com o que foi solicitado, o simulador deve ser desenvolvido exclusivamente em **C** e irá implementar a política de escalonamento **Round Robin com Feedback**, que combina a execução circular típica do Round Robin com uma análise dinâmica de prioridades (Feedback). Essa análise ajusta os processos entre filas de diferentes níveis de prioridade, com base em regras específicas, como solicitações de entrada/saída ou preempção.

Exemplo de saída

A saída do simulador mostra, a cada 2 segundos (sleep), as iterações do simulador. Podemos ver tanto o tempo atual (o quanto já passou) como todas as filas e suas mudanças. Assim, conseguimos saber exatamente em que fila cada processo está. Quando um processo termina de ser executado, ele vai para a fila de concluídos, que também é mostrada.

```
TEMPO = 9

Fila Finalizados:
[ 3 ]
Fila Alta Prioridade:

Fila Baixa Prioridade:
[ 1 ] [ 4 ] [ 5 ]
Fila Disco:
[ 2 ]
Fila Fita:

Fila Impressora:

Fila Bloqueados:
[ 2 ]
IO(chegada=2, tipo=3, tempo_utilizado=3, PID=2)
```

A imagem acima mostra a saída do simulador quando ele está em 9 u.t.; O Processo 3 foi concluído em 8 u.t. e movido para a fila de finalizados. O Processo 5 chegou em 8 u.t. e em 9 u.t. desceu para baixa prioridade pois não concluiu seu processamento a tempo. Os Processos 1 e 4 também estão em baixa prioridade, enquanto o Processo 2 está bloqueado, pois foi chamado em 7 u.t. por Disco e precisa ficar aguardando 3 u.t. até ser liberado, que é a fatia de tempo dessa fila.

Metodologia de Implementação

Premissas Consideradas

Abaixo, é possível visualizar todas as premissas usadas para o trabalho, que serão detalhadas ao longo do relatório.

- Número de processos: **5**
- Fatia de tempo (Quantum): **1 u.t.**
- Tempos para cada tipo de I/O:
 - ◆ Disco: **3 u.t.**
 - ◆ Fita magnética: **4 u.t.**
 - ◆ Impressora: **5 u.t.**
- Filas de CPU:
 - ◆ 1 fila de prioridade **alta**
 - ◆ 1 fila de prioridade **baixa**
- Filas de I/O:
 - ◆ 1 fila para operações de **disco**, onde o processo retorna para a fila de **baixa** prioridade quando seu tempo é concluído
 - ◆ 1 fila para operações de **fita magnética**, onde o processo retorna para a fila de **alta** prioridade quando seu tempo é concluído
 - ◆ 1 fila para operações de **impressora**, onde o processo retorna para a fila de **alta** prioridade quando seu tempo é concluído
- Gerenciamento:
 - ◆ Todos os processos têm PID, que servem como **identificação** no simulador
 - ◆ Todos os processos têm um tempo de serviço e I/O **fixos**
 - ◆ Processos novos sempre entram na fila de **alta prioridade**
 - ◆ Processos que sofrem preempção são movidos para a fila de **baixa prioridade**

Tabela de Processos

| | PID | Tempo de chegada | Tempo de serviço | Tempo de I/O | Tipo de I/O |
|-------------|-----|------------------|------------------|--------------|-------------|
| Processo #1 | 1 | 0 | 8 | 4 | Disco |
| Processo #2 | 2 | 2 | 6 | 2, 6 | Disco, Fita |
| Processo #3 | 3 | 3 | 2 | - | - |
| Processo #4 | 4 | 5 | 3 | - | - |
| Processo #5 | 5 | 8 | 11 | 5 | Impressora |

Algoritmo de Escalonamento

No algoritmo **Round Robin com Feedback**, o tempo de CPU é distribuído de forma igual entre os processos por meio de fatias de tempo fixas (**1 u.t.**, como escolhido no nosso trabalho). Após consumir esse tempo, caso o processo **não termine** sua execução, ele será **preemptado**. Quando isso acontecer, se ele estiver em uma fila de alta prioridade, será movido para uma fila de baixa prioridade. Caso já esteja na fila de baixa prioridade, ele continuará nela.

Durante a execução do simulador, um processo pode ser **solicitado** por um dispositivo I/O, sendo movido para sua fila correspondente. Todas as filas de I/O têm um tempo fixo onde os processos devem permanecer aguardando (em espera). Quando finalizado, eles podem voltar para as filas de alta ou baixa prioridade de CPU, dependendo do **tipo de dispositivo** que o solicitou antes.

O processo é liberado quando tanto seu tempo de CPU quanto seu tempo de I/O são zerados, fazendo com que ele saia de todas as filas e complete o seu serviço.

Estrutura de Filas e Gerenciamento de I/O

Filas de Prioridade

- **Fila de Alta Prioridade:** Contém processos recém-chegados e aqueles retornados de operações de I/O de alta prioridade (fita magnética ou impressora). Esses processos têm preferência para a execução na CPU, recebendo prioridade máxima.
- **Fila de Baixa Prioridade:** Processos que foram preemptados por terem consumido toda a sua fatia de tempo ou que retornaram de operações de I/O de baixa prioridade (disco) são encaminhados para esta fila. Eles têm menor prioridade e serão atendidos após os processos da fila de alta prioridade.

Filas de I/O

Para simular o comportamento dos dispositivos, utilizam-se **filas separadas para diferentes tipos de I/O**. Com isso, temos três filas de I/O: Disco, Fita Magnética e Impressora, onde cada uma tem seus próprios tempos e especificações.

- **Disco:** O processo fica 3 u.t. aguardando na fila e após concluir a operação, retorna à fila de baixa prioridade.
- **Fita Magnética:** Fica aguardando por 4 u.t. e retorna à fila de alta prioridade.
- **Impressora:** Fica aguardando por 5 u.t. e retorna à fila de alta prioridade, assim como a fila de fita magnética.

Cada fila de I/O é gerenciada de forma independente, com tempos fixos para completar as operações. Durante o tempo de espera, o escalonador continua atendendo as filas de alta e baixa prioridade.

Implementação

Detalhes de Desenvolvimento

Inicialmente, durante o desenvolvimento do escalonador, houveram algumas dificuldades em verificar o andamento dos processos. Para solucionar esse problema foram feitas algumas escolhas, sendo elas:

- **Fatia de tempo:** 1 u.t
- **Tempo Disco:** 3 u.t
- **Tempo Fita magnética:** 4 u.t
- **Tempo Impressora:** 5 u.t

O critério usado para definir o valor da fatia de tempo foi a **visualização**. Uma fatia de tempo menor facilitou o controle sobre o comportamento dos processos e garantiu uma execução **mais previsível** durante os testes. Os tempos de I/O foram escolhidos dessa maneira para facilitar a implementação e integração com as filas.

Foram definidas **5 filas**: 1 fila de alta prioridade, 1 fila de baixa prioridade, 1 fila de disco, 1 fila de fita magnética e 1 fila de impressora. Os processos são lidos de um arquivo.txt e os tipos foram enumerados como 3, 4, 5, que por sua vez também são os tempos dos I/O. Isso não é por acaso - foram definidas dessa forma a fim de tornar a implementação mais intuitiva e consistente.

O controle dos PIDs é feito a partir de uma variável global. Sempre que um processo começa é definido um PID para ele incrementando em +1 a variável global responsável.

Resultados

Ordem de Finalizações: [3, 4, 1, 2, 5]

Saídas do Simulador: Verificar no [Apêndice A](#)

Saídas Detalhadas:

- 0 u.t.
 - ◆ **P1** chega na CPU, em fila de alta prioridade. Faltam: 8 u.t.
- 1 u.t.
 - ◆ **P1** não termina sua execução a tempo e é movido para a fila de baixa prioridade. Faltam: 7 u.t.
- 2 u.t.
 - ◆ **P1** continua na fila de baixa prioridade. Faltam: 6 u.t.
 - ◆ **P2** chega na CPU, em fila de alta prioridade. Faltam: 6 u.t.
- 3 u.t.
 - ◆ **P2** não termina sua execução a tempo e é movido para a fila de baixa prioridade. Faltam: 5 u.t.
 - ◆ **P3** chega na CPU, em fila de alta prioridade. Faltam: 2 u.t.
- 4 u.t.
 - ◆ **P3** não termina sua execução a tempo e é movido para a fila de baixa prioridade. Faltam: 1 u.t.
- 5 u.t.
 - ◆ **P1** não termina sua execução a tempo e continua na fila de baixa prioridade. Faltam: 5 u.t.
 - ◆ **P4** chega na CPU, em fila de alta prioridade. Faltam: 3 u.t.
- 6 u.t.
 - ◆ **P4** não termina sua execução a tempo e é movido para a fila de baixa prioridade. Faltam: 2 u.t.
- 7 u.t.
 - ◆ **P2** não termina sua execução a tempo. Faltam: 4 u.t.
 - ◆ **P2** é chamado por Disco, pois foi executado por 2 u.t. Precisa ficar em Disco por 3 u.t. Faltam: 3 u.t.
- 8 u.t.
 - ◆ **P3** é finalizado.

- ◆ **P2** ainda precisa aguardar em Disco. Faltam: 2 u.t.
- ◆ **P5** chega na CPU, em fila de alta prioridade. Faltam: 11 u.t.
- **9 u.t.**
 - ◆ **P5** não termina sua execução a tempo e é movido para a fila de baixa prioridade. Faltam: 10 u.t.
 - ◆ **P2** ainda precisa aguardar em Disco. Faltam: 1 u.t.
- **10 u.t.**
 - ◆ **P1** não termina sua execução a tempo. Faltam: 4 u.t.
 - ◆ **P1** é chamado por Disco, pois foi executado por 4 u.t. Precisa ficar em Disco por 3 u.t. Faltam: 3 u.t.
 - ◆ **P2** é liberado por Disco e volta para a fila de baixa prioridade. Faltam: 4 u.t.
- **11 u.t.**
 - ◆ **P4** não termina sua execução a tempo. Faltam: 1 u.t.
 - ◆ **P1** ainda precisa aguardar em disco. Faltam: 2 u.t.
- **12 u.t.**
 - ◆ **P5** não termina sua execução a tempo. Faltam: 9 u.t.
 - ◆ **P1** ainda precisa aguardar em disco. Faltam: 1 u.t.
- **13 u.t.**
 - ◆ **P1** é liberado por Disco e volta para a fila de baixa prioridade. Faltam: 4 u.t.
 - ◆ **P2** não termina sua execução a tempo. Faltam: 3 u.t.
- **14 u.t.**
 - ◆ **P4 é finalizado.**
- **15 u.t.**
 - ◆ **P5** não termina sua execução a tempo. Faltam: 8 u.t.
- **16 u.t.**
 - ◆ **P1** não termina sua execução a tempo. Faltam: 3 u.t.
- **17 u.t.**
 - ◆ **P2** não termina sua execução a tempo. Faltam: 3 u.t.
- **18 u.t.**
 - ◆ **P5** não termina sua execução a tempo. Faltam: 7 u.t.
- **19 u.t.**
 - ◆ **P1** não termina sua execução a tempo. Faltam: 2 u.t.
- **20 u.t.**
 - ◆ **P2** não termina sua execução a tempo. Faltam: 2 u.t.
- **21 u.t.**

- ◆ **P5** não termina sua execução a tempo. Faltam: 6 u.t.
- ◆ **P5** é chamado por Impressora, pois foi executado por 5 u.t. Precisa ficar em Impressora por 5 u.t. Faltam: 5 u.t.
- **22 u.t.**
 - ◆ **P1** não termina sua execução a tempo. Faltam: 1 u.t.
 - ◆ **P5** ainda precisa aguardar em Impressora. Faltam: 4 u.t.
- **23 u.t.**
 - ◆ **P2** não termina sua execução a tempo. Faltam: 1 u.t.
 - ◆ **P2** é chamado por Fita, pois foi executado por 6 u.t. Precisa ficar em fita por 4 u.t. Faltam: 4 u.t.
 - ◆ **P5** ainda precisa aguardar em Impressora. Faltam: 3 u.t.
- **24 u.t.**
 - ◆ **P1** é finalizado.
 - ◆ **P2** ainda precisa aguardar em Fita. Faltam: 3 u.t.
 - ◆ **P5** ainda precisa aguardar em Impressora. Faltam: 2 u.t.
- **25 u.t.**
 - ◆ **P2** ainda precisa aguardar em Fita. Faltam: 2 u.t.
 - ◆ **P5** ainda precisa aguardar em Impressora. Faltam: 1 u.t.
- **26 u.t.**
 - ◆ **P2** ainda precisa aguardar em Fita. Faltam: 1 u.t.
 - ◆ **P5** é liberado por Impressora e volta para a fila de alta prioridade. Faltam: 6 u.t.
- **27 u.t.**
 - ◆ **P2** é liberado por Fita e volta para a fila de alta prioridade. Faltam: 1 u.t.
 - ◆ **P5** não termina sua execução a tempo e é movido para a fila de baixa prioridade. Faltam: 5 u.t.
- **28 u.t.**
 - ◆ **P2** é finalizado.
 - .
. *// como só sobrou P5, ele vai rodar sozinho até acabar*
.
- **33 u.t.**
 - ◆ **P5** é finalizado.

Discussões dos Resultados

Análise do Comportamento do Escalonador

Com um intervalo de *sleep* de dois segundos por iteração, o algoritmo conclui todos os processos em aproximadamente **1 minuto e 10 segundos**. Considerando que os tempos de serviço e as chamadas de I/O dos processos são fixos, de acordo com as premissas definidas, o escalonador opera por um total de **33 unidades de tempo (u.t.)**. Esse é o tempo total necessário para finalizar a execução, utilizando uma fatia de tempo de **1 u.t.** para cada operação.

Observações e Dificuldades

A primeira versão do simulador conseguiu gerir bem tanto Round Robin quanto Feedback, mas não conseguimos implementar perfeitamente a parte de I/O de forma paralela. O maior problema foram as dúvidas iniciais de como realmente funcionava ou o que era esperado de acontecer, além de manter consistência na hora de rodar as fatias de tempo. Eventualmente, decidimos fazer um modelo onde consideramos a fatia de tempo como 1 u.t. para sua implementação ser mais simples.

Divisão do trabalho

Durante o desenvolvimento do trabalho, todos os membros participaram de todas as suas partes, tanto em código como no relatório.

No que se refere ao código, podemos dividir em 2 passos o desenvolvimento: Primeiro passo, a construção de um escalonador sem I/O. Essa primeira parte foi feita pela integrante Bruna. A segunda parte, o desenvolvimento do gerenciamento do I/O foi feita pelos integrantes Lucas e Eduardo. O relatório também foi um trabalho em conjunto feito por todos os integrantes.

Conclusão

Principais Pontos

O trabalho permitiu compreender o funcionamento do algoritmo Round Robin com Feedback, mostrando como ele distribui o tempo de CPU de forma justa e ajusta as prioridades dos processos de acordo com suas necessidades. Isso destacou a importância do escalonamento para otimizar o uso da CPU e reduzir tempos de espera.

A simulação também revelou desafios, como o gerenciamento de filas de prioridade e operações de I/O em paralelo, reforçando a necessidade de estratégias eficientes para lidar com múltiplos processos em sistemas operacionais.

Apesar do simulador não estar preparado para lidar com fatias de tempo maiores, completou seu objetivo de forma satisfatória, que é ajudar a entender melhor como um escalonador funciona.

Possíveis Melhorias e Trabalhos Futuros

Um processo pode ter zero, uma ou várias chamadas de I/O, o que é mais realista do que sempre manter uma chamada por processo, por exemplo. Essa implementação ajudou bastante a entender um pouco melhor como funcionaria em um sistema mais realista, onde nem sempre todos os processos terão as mesmas características.

Gostaríamos de pesquisar as mudanças necessárias para poder testar entre outras fatias de tempo e ver o funcionamento do programa. Optamos por manter a fatia de tempo em 1 u.t., pois facilita os testes e a análise dos resultados. Essa escolha, no entanto, limita os valores dos processos, que precisam ser pequenos para evitar que a simulação demore. Por enquanto, não precisamos lidar com chamadas de I/O acontecendo no meio de uma fatia de tempo e suas implicações, o que torna o código mais simples de ser implementado, mas menos ideal.

Referências

- > **Valéria M. Bastos**, Escalonamento de Processos, 2021. Disponível em: <https://drive.google.com/file/d/1gRw99xsFZIFRwB6mhn8iaOUhPVugNCSE/view>. Acesso em: 09 nov. 2024.
- > **Islene Calciolari Garcia**, Escalonamento de Processos e Threads, 2008. Disponível em: <https://www.ic.unicamp.br/~islene/1s2008-mc514/sched/sched.pdf>. Acesso em: 11 nov. 2024.
- > **DEINFO**, ROUND ROBIN, uma técnica preemptiva de escalonamento, 2016. Disponível em: <https://deinfo.uepg.br/~alunoso/2016/ROUNDROBIN/>. Acesso em: 11 nov.
- > **Prashant Shenoy**, Operating Systems, 2012. Disponível em: https://lass.cs.umass.edu/~shenoy/courses/fall12/lectures/notes/Lec05_notes.pdf. Acesso em: 15 nov.
- > **GEEKSFORGEEKS**, Round Robin Scheduling with different arrival times, 2024. Disponível em: <https://www.geeksforgeeks.org/round-robin-scheduling-with-different-arrival-times/>. Acesso em: 15 nov.
- > **JAVATPOINT**, Round Robin Program in C with Output. Disponível em: <https://www.javatpoint.com/round-robin-program-in-c>. Acesso em: 15 nov.

Apêndice

Apêndice A - Saída dos Resultados

| | | |
|---|---|---|
| <pre>TEMPO = 0 Fila Finalizados: Fila Alta Prioridade: [1] Fila Baixa Prioridade: Fila Disco: Fila Fita: Fila Impressora: Fila Bloqueados:</pre> | <pre>TEMPO = 3 Fila Finalizados: Fila Alta Prioridade: [3] Fila Baixa Prioridade: [1] [2] Fila Disco: Fila Fita: Fila Impressora: Fila Bloqueados:</pre> | <pre>TEMPO = 6 Fila Finalizados: Fila Alta Prioridade: Fila Baixa Prioridade: [2] [3] [1] [4] Fila Disco: Fila Fita: Fila Impressora: Fila Bloqueados:</pre> |
| <pre>TEMPO = 1 Fila Finalizados: Fila Alta Prioridade: Fila Baixa Prioridade: [1] Fila Disco: Fila Fita: Fila Impressora: Fila Bloqueados:</pre> | <pre>TEMPO = 4 Fila Finalizados: Fila Alta Prioridade: Fila Baixa Prioridade: [1] [2] [3] Fila Disco: Fila Fita: Fila Impressora: Fila Bloqueados:</pre> | <pre>TEMPO = 7 Fila Finalizados: Fila Alta Prioridade: Fila Baixa Prioridade: [3] [1] [4] Fila Disco: [2] Fila Fita: Fila Impressora: Fila Bloqueados: [2] IO(chegada=2, tipo=3, tempo_utilizado=1, PID=2)</pre> |
| <pre>TEMPO = 2 Fila Finalizados: Fila Alta Prioridade: [2] Fila Baixa Prioridade: [1] Fila Disco: Fila Fita: Fila Impressora: Fila Bloqueados:</pre> | <pre>TEMPO = 5 Fila Finalizados: Fila Alta Prioridade: [4] Fila Baixa Prioridade: [2] [3] [1] Fila Disco: Fila Fita: Fila Impressora: Fila Bloqueados:</pre> | <pre>TEMPO = 8 Fila Finalizados: [3] Fila Alta Prioridade: [5] Fila Baixa Prioridade: [1] [4] Fila Disco: [2] Fila Fita: Fila Impressora: Fila Bloqueados: [2] IO(chegada=2, tipo=3, tempo_utilizado=2, PID=2)</pre> |

TEMPO = 9

Fila Finalizados:
[3]
Fila Alta Prioridade:

Fila Baixa Prioridade:
[1] [4] [5]
Fila Disco:
[2]
Fila Fita:

Fila Impressora:

Fila Bloqueados:
[2]
IO(chegada=2, tipo=3, tempo_utilizado=3, PID=2)

TEMPO = 12

Fila Finalizados:
[3]
Fila Alta Prioridade:

Fila Baixa Prioridade:
[2] [4] [5]
Fila Disco:
[1]
Fila Fita:

Fila Impressora:

Fila Bloqueados:
[1]
IO(chegada=4, tipo=3, tempo_utilizado=3, PID=1)

TEMPO = 15

Fila Finalizados:
[3] [4]
Fila Alta Prioridade:

Fila Baixa Prioridade:
[1] [2] [5]
Fila Disco:

Fila Fita:

Fila Impressora:

Fila Bloqueados:

TEMPO = 10

Fila Finalizados:
[3]
Fila Alta Prioridade:

Fila Baixa Prioridade:
[4] [5] [2]
Fila Disco:
[1]
Fila Fita:

Fila Impressora:

Fila Bloqueados:
[1]
IO(chegada=4, tipo=3, tempo_utilizado=1, PID=1)

TEMPO = 13

Fila Finalizados:
[3]
Fila Alta Prioridade:

Fila Baixa Prioridade:
[4] [5] [1] [2]
Fila Disco:

Fila Fita:

Fila Impressora:

Fila Bloqueados:

TEMPO = 16

Fila Finalizados:
[3] [4]
Fila Alta Prioridade:

Fila Baixa Prioridade:
[2] [5] [1]
Fila Disco:

Fila Fita:

Fila Impressora:

Fila Bloqueados:

TEMPO = 11

Fila Finalizados:
[3]
Fila Alta Prioridade:

Fila Baixa Prioridade:
[5] [2] [4]
Fila Disco:
[1]
Fila Fita:

Fila Impressora:

Fila Bloqueados:
[1]
IO(chegada=4, tipo=3, tempo_utilizado=2, PID=1)

TEMPO = 14

Fila Finalizados:
[3] [4]
Fila Alta Prioridade:

Fila Baixa Prioridade:
[5] [1] [2]
Fila Disco:

Fila Fita:

Fila Impressora:

Fila Bloqueados:

TEMPO = 17

Fila Finalizados:
[3] [4]
Fila Alta Prioridade:

Fila Baixa Prioridade:
[5] [1] [2]
Fila Disco:

Fila Fita:

Fila Impressora:

Fila Bloqueados:

TEMPO = 18

Fila Finalizados:

[3] [4]

Fila Alta Prioridade:

Fila Baixa Prioridade:

[1] [2] [5]

Fila Disco:

Fila Fita:

Fila Impressora:

Fila Bloqueados:

TEMPO = 21

Fila Finalizados:

[3] [4]

Fila Alta Prioridade:

Fila Baixa Prioridade:

[1] [2]

Fila Disco:

Fila Fita:

Fila Impressora:

[5]

Fila Bloqueados:

[5]

TEMPO = 24

Fila Finalizados:

[3] [4] [1]

Fila Alta Prioridade:

Fila Baixa Prioridade:

Fila Disco:

Fila Fita:

[2]

Fila Impressora:

[5]

Fila Bloqueados:

[5] [2]

—

TEMPO = 27

Fila Finalizados:
[3] [4] [1]
Fila Alta Prioridade:
[2]
Fila Baixa Prioridade:
[5]
Fila Disco:

Fila Fita:

Fila Impressora:

Fila Bloqueados:

TEMPO = 30

Fila Finalizados:
[3] [4] [1] [2]
Fila Alta Prioridade:
Fila Baixa Prioridade:
[5]
Fila Disco:

Fila Fita:

Fila Impressora:

Fila Bloqueados:

TEMPO = 33

Fila Finalizados:
[3] [4] [1] [2] [5]
Fila Alta Prioridade:
Fila Baixa Prioridade:
Fila Disco:

Fila Fita:

Fila Impressora:

Fila Bloqueados:

TEMPO = 28

Fila Finalizados:
[3] [4] [1] [2]
Fila Alta Prioridade:

Fila Baixa Prioridade:
[5]
Fila Disco:

Fila Fita:

Fila Impressora:

Fila Bloqueados:

TEMPO = 31

Fila Finalizados:
[3] [4] [1] [2]
Fila Alta Prioridade:

Fila Baixa Prioridade:
[5]
Fila Disco:

Fila Fita:

Fila Impressora:

Fila Bloqueados:

TEMPO = 29

Fila Finalizados:
[3] [4] [1] [2]
Fila Alta Prioridade:

Fila Baixa Prioridade:
[5]
Fila Disco:

Fila Fita:

Fila Impressora:

Fila Bloqueados:

TEMPO = 32

Fila Finalizados:
[3] [4] [1] [2]
Fila Alta Prioridade:

Fila Baixa Prioridade:
[5]
Fila Disco:

Fila Fita:

Fila Impressora:

Fila Bloqueados: