FATEC

CENTRO PAULA SOUZA
COMPETÊNCIA EM EDUCAÇÃO PÚBLICA PROFISSIONAL
Faculdade de Tecnologia Prof. Jessen Vidal
São José dos Campos

GOVERNO DO ESTADO DE SÃO PAULO

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Disciplina: Sistemas Operacionais I

Aula 16: Escalonamento de CPU P2

Prof. Diogo Branquinho Ramos
diogo.branquinho@fatec.sp.gov.br
São José dos Campos - SP

Roteiro

- Algoritmos de escalonamento
- Escalonamento em múltiplos processadores
- Balanceamento de carga

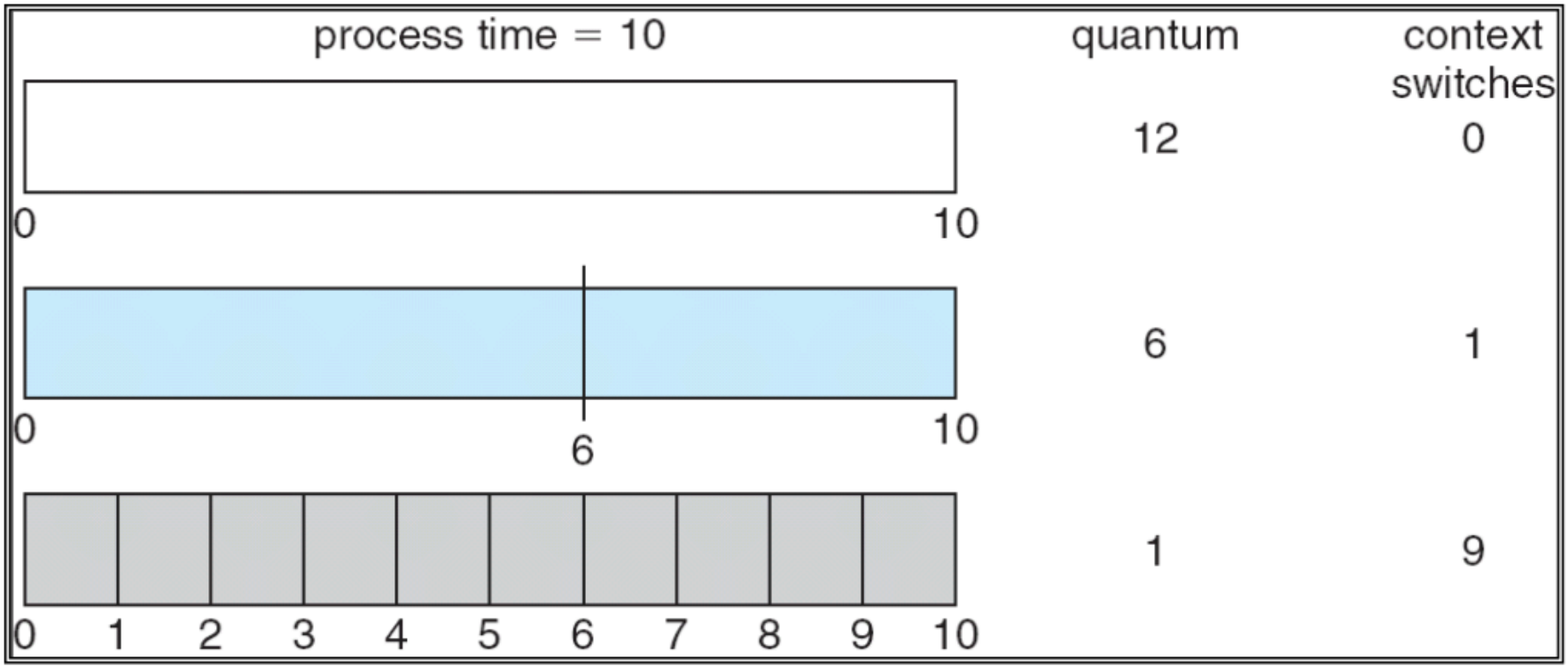
Round Robin (RR)

- **Quantum de tempo**
  - Define o tempo máximo de uso da CPU por aquele processo (normalmente 10-100 milissegundos).
- **Escalonamento circular: FIFO circular**
- **Preemptivo**
  - Acabou o quantum de tempo, o processo sai de contexto.
- **Funcionamento**
  - Se houver n processos na fila de prontos (circular) e o quantum de tempo for q, então cada processo recebe 1/n do tempo de CPU em pedaços de no máximo q unidades de tempo de uma só vez. Nenhum processo espera mais do que (n – 1)q unidades de tempo.

Round Robin (RR)

- **Desempenho**
  - Depende do tamanho do quantum de tempo
  - q grande ➔ FIFO (política FCFS).
  - q pequeno ➔ A troca de contexto começa a influenciar.
  - q deve ser grande em relação à troca de contexto, ou então o overhead é muito alto.
  - A troca de contexto pode durar 10 microssegundos.
  - 80% dos bursts de CPU devem ser menores que o quantum de tempo.

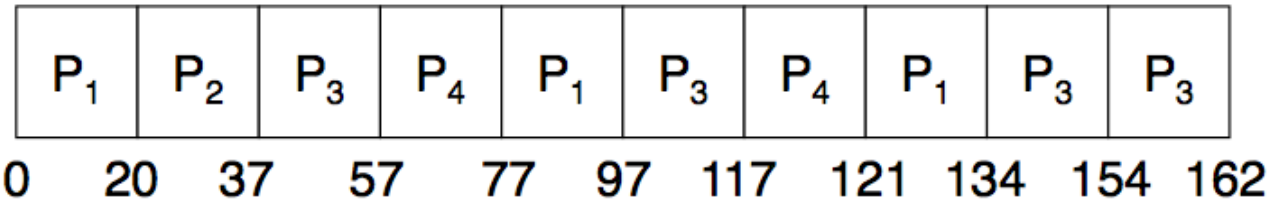
Round Robin (RR)



Round Robin (RR) – quantum = 20

Processo	Tempo de burst
P1	53
P2	17
P3	68
P4	24

O diagrama de Gantt é:



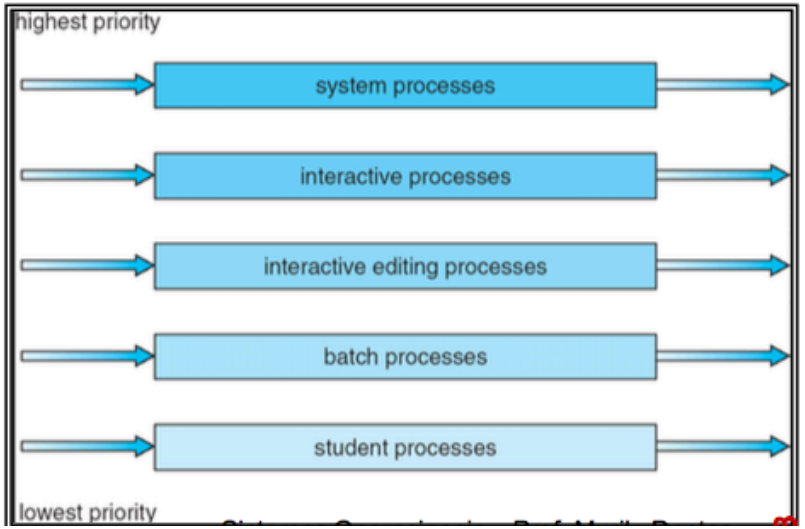
Normalmente, maior turnaround médio que SJF, porém com resposta melhor!

Fila Multinível

- **Divisão da fila de prontos (critério)**
  - Ex.: primeiro plano (interativo) e segundo plano (batch).
    - Possuem requisitos diferentes para tempo de resposta.
      - Memória, prioridade, tipo de processo...
- **Cada fila tem seu próprio algoritmo de escalonamento**
  - Ex.:
    - Primeiro plano – RR.
    - Segundo plano – FCFS.
- **Escalonamento entre as filas**
  - Escalonamento com prioridade fixa
    - Serve tudo em primeiro plano, depois em segundo plano.
    - Possibilidade de estagnação.

Fila Multinível

- **Escalonamento entre as filas (outra opção)**
  - Fatia de tempo
    - Cada fila recebe uma certa quantidade de tempo de CPU, que ela pode escalonar entre seus processos.
      - 80% para primeiro plano no RR.
      - 20% para segundo plano no FCFS.



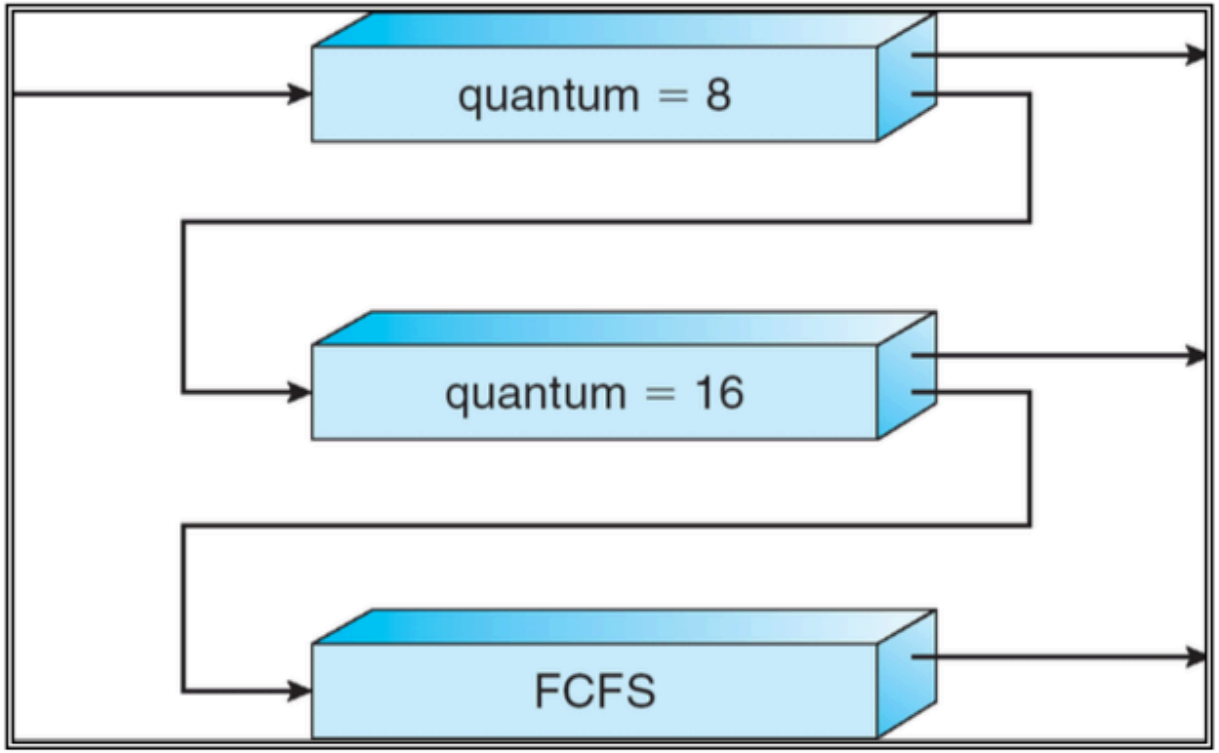
Fila Multinível com Feedback

- **Novidade: um processo pode se mover entre as diversas filas**
  - O envelhecimento pode ser implementado dessa forma: mover o processo para uma fila de maior prioridade.
- **Parâmetros do escalonador:**
  - Número de filas.
  - Algoritmos de escalonamento para cada fila.
  - Método usado para determinar quando fazer o upgrade de um processo.
  - Método usado para determinar quando rebaixar um processo.
  - Método usado para determinar em qual fila um processo entrará quando esse processo precisar de serviço.

Exemplo de Fila Multinível com Feedback

- **Três filas:**
  - Q0 – RR com quantum de tempo de 8 ms.
  - Q1 – RR com quantum de tempo de 16 ms.
  - Q2 – FCFS.
- **Escalonamento**
  - O job entra na fila Q0. Quando ganha a CPU, o job recebe 8 ms. Se não terminar em 8 ms, é movido para a fila Q1.
  - Em Q1 o job recebe 16 ms adicionais. Se não completar, ele é substituído e movido para a fila Q2.

Exemplo de Fila Multinível com Feedback



Escalonamento de múltiplos processadores

- **Escalonamento mais complexo**
  - Não existe melhor solução!
- **Mesmo com CPUs homogêneas!**
  - Arquitetura pode complicar o escalonamento.
    - Ex.: E/S ligado a um barramento exclusivo de um dos processadores: processo usando tal E/S precisa ser escalonado neste processador.

Escalonamento de múltiplos processadores

- **Afinidade de processador**
  - A mudança entre processadores pode aumentar o custo computacional.
    - Invalidação e repreenchimento de caches.
- **Afinidade rígida**
  - Não permite a migração de processos para outros processadores.
- **Afinidade flexível**
  - Não garante a execução do processo no mesmo processador até o final.

Balanceamento de carga

- **Funcionamento**
  - Mantém uniformidade na carga de trabalho entre as CPUs.
- **Técnicas**
  - Migração push
    - Uma rotina verifica periodicamente a carga de cada CPU. Se encontrar desequilíbrio, empurra processos de uma CPU para outra.
  - Migração pull
    - Um processador ocioso puxa tarefas esperando pelo outro processador para si.
  - Não são mutuamente exclusivas.
    - São implementadas em paralelo em muitos SOs.
  - Invalida o benefício da afinidade.

ESCALONAMENTO DO LINUX

Organização básica

- **Tipos de processo**
  - Interativos
  - Batch
  - Tempo Real
    - O escalonador não distingue interativos de batch, mas os distingue de tempo real.
- **Subtipos: CPU-bound e I/O-bound**
  - Como todos os UNIX, privilegia I/O-bound.
    - Fornece melhor tempo de resposta para aplicações interativas.
- **Escalonador baseado em time-sharing.**
  - O escalonamento é preemptivo.

Características do escalonador

- **Cada processo recebe um quantum no momento da criação**
  - Diferentes processos podem possuir diferentes valores de quantum.
  - O quantum é um múltiplo de 10 ms, inferior a 100 ms.
- **O escalonador executa primeiro os processos de tempo real e só assim executa os demais de acordo com a prioridade.**
- **fork-exec**
  - Quando um p-pai cria um p-filho, ele cede metade de seu quantum restante ao p-filho por questões de segurança.

Características do escalonador

- **Prioridades**
  - Dinâmicas: mudam em tempo de execução (de 100 a 199).
    - Processo que usou muito a CPU tem sua prioridade reduzida; ou
    - Processo há muito tempo sem usar a CPU, a prioridade sobe.
    - Aplicada em processos batch e interativos.
      - A prioridade é calculada pela prioridade base e o tempo restante do quantum.
  - Estáticas: exclusivas para processos de tempo real.
    - Varia de 0-99.
    - Definida pelo usuário.
    - Não é modificada pelo escalonador.
    - Só usuários com privilégios especiais podem criar e definir processos em tempo real.

Políticas de escalonamento

- **SCHED_FIFO**
  - Processo de tempo real em fila de acordo com sua prioridade.
  - Liberação da CPU: espontânea, ou na chegada de outro processo com prioridade maior, ou no bloqueio devido a I/O, ou através de sincronismo.
- **SCHED_RR**
  - Também usada com processos de tempo real.
  - Liberação da CPU: idem anterior, com a possibilidade também de fim do quantum (5 possibilidades).
- **SCHED_OTHER**
  - Usado para processos interativos e batch.
  - Filas de Multinível de prioridades dinâmicas com timesharing.

ESCALONAMENTO DO WINDOWS

Organização básica

- **Usa um esquema de 32 níveis de prioridades**
  - Tempo real: 16 a 31 (prioridade fixa)
  - Demais: 0 a 15 (prioridade dinâmica)
- **Estratégia**
  - Melhores tempos para threads interativos.
  - Permite que threads I/O-bound mantenham os dispositivos em funcionamento.
  - Threads CPU-bound ficam com a sobra de CPU num contexto de background.
- **Threads de tempo real tem preferência**
- **Quanto maior o inteiro, maior a prioridade**

Organização básica

- **Usa quatro classes**
  - IDLE_PRIORITY_CLASS (nível 4)
  - NORMAL_PRIORITY_CLASS (nível 8 - a mais comum)
  - HIGH_PRIORITY_CLASS (nível 13)
  - REALTIME_PRIORITY_CLASS (nível 24)
- **Distinção entre processo foreground (o selecionado) e o background (os demais)**
  - Quando um processo background se torna foreground, sua prioridade aumenta em 3.
- **Cada prioridade possui uma fila e cada fila é atendida por RR**
  - Um processo de prioridade dinâmica migra entre as filas.