Trabalho Prático #2 – Escalonador de Processos

Joao Pedro Pinto Coelho <jppcsc96@gmail.com>

Rafael Augusto Botelho Perez <rafael.perez@dcc.ufmg.br>

Termo de compromisso

Os membros do grupo afirmam que todo o código desenvolvido para este trabalho é de autoria própria. Exceto pelo material listado no item 3 deste relatório, os membros do grupo afirmam não ter copiado material da Internet nem obtiveram código de terceiros.

Membros do grupo e alocação de esforço

Preencha as linhas abaixo com o nome e o e-mail dos integrantes do grupo. Substitua XX pela contribuição de cada membro do grupo no desenvolvimento do trabalho.

- Joao Pedro Pinto Coelho <jppcsc96@gmail.com> 50%
- Rafael Augusto Botelho Perez <rafael.perez@dcc.ufmg.br> 50%

Referências bibliográficas

COX, Russ. “xv6: a simple, Unix-like teaching operating system”. <https://pdos.csail.mit.edu/6.828/2018/xv6/book-rev11.pdf>. Acesso em 30 set. 2019.

YU, Dr. Tong Lai. “Writing a Simple Shell and XV6 Scheduling and System calls”. Disponível em: <http://cse.csusb.edu/tongyu/courses/cs660/labs/lab3.php>. Acesso em 30 set. 2019.

Escalonador

1. **Qual a política de escalonamento é utilizada atualmente no XV6?**

   Round-robin (RR).

2. **Quais processos essa política seleciona para rodar?**

   A ideia é que um processo de alta prioridade executável seja preferido pelo agendador em vez de um processo de baixa prioridade executável.

3. **O que acontece quando um processo retorna de uma tarefa de I/O?**

   O Xv6 multiplexa alternando cada processador de um processo para outro em duas situações. Primeiro, o mecanismo de `sleep` e `wake` do xv6 muda quando um processo espera que a I/O do dispositivo ou pipe seja concluída, ou espera que uma child termine, ou espera na chamada do sistema de sono. Em segundo lugar, o xv6 periodicamente força um switch quando um processo está executando instruções do usuário. Esta multiplexação cria a ilusão de que cada processo tem sua própria CPU, assim como o xv6 usa o alocador de memória e tabelas de página de hardware para criar a ilusão que cada processo tem a sua própria memória.

4. **O que acontece quando um processo é criado e quando ou quão frequente o escalonamento acontece?**

   Quando um processo é criado, ele recebe a CPU por um tempo pequeno (quantum). O escalonador do xv6 implementa uma política de escalonamento

simples, que executa cada processo por sua vez. Esta política é chamada de Round Robin. Sistemas operacionais reais implementam políticas mais sofisticadas que, por exemplo, permitem que os processos tenham prioridades. A idéia é que um processo de alta prioridade executável seja preferido pelo agendador em vez de um processo de baixa prioridade executável. Essas políticas podem se tornar complexas rapidamente porque muitas vezes existem objetivos concorrentes: por exemplo, a operação também pode querer garantir justiça e alto rendimento. Além disso, políticas complexas podem levar a interações não intencionais, como inversão de prioridades e comboios. A inversão de prioridades pode acontecer quando um processo de baixa prioridade e alta prioridade compartilham um bloqueio, que, quando adquirido pelo processo de baixa prioridade, pode impedir que o processo de alta prioridade progrida. Um comboio longo pode formar-se quando muitos processos de alta prioridade estão à espera de um processo de baixa prioridade que adquire um bloqueio partilhado; uma vez formado, o comboio pode persistir por muito tempo. Para evitar esses tipos de problemas, mecanismos adicionais são necessários em programadores sofisticados.

Algoritmos implementados

Round Robin (RR). Cada processo recebe a CPU por um tempo pequeno (quantum). Usualmente, de 10 a 100 milissegundos. Ao fim do tempo, processo é preemptado e vai para a fim da fila de prontos. Preempção periódica é um overhead.

- quantum muito grande: FIFO

- quantum muito pequeno: overhead de preempção se torna proibitivo

Escalonamento preemptivo. Periodicamente o escalonador interrompe o processo em execução e muda-o para “pronto”. Escalonador mais complexo, compartilhamento da CPU é garantido

Escalonamento (filas) multi-nível

- Fila de prontos é dividida:
    - processos interativos (foreground) – RR
    - processos em lote/batch (background) – FCFS
- Escalonamento entre filas
    - Prioridade fixa. P.ex.: atender sempre processo interativos primeiro
    - Possibilidade de inanição
- Fatias de tempo. Cada fila é escalonada por uma fração do tempo total

Análise dos resultados dos testes obtidos

Dentre as tarefas implementadas, o grupo não pode concluir os testes.