

### Descrição/Especificação dos Trabalhos I e II

**Objetivo geral:** desenvolver conhecimento e prática de implementação de um componente básico (i.e., escalonador de processos) em um sistema operacional didático.

**Objetivos específicos:** **a)** estudar o código de uma implementação do Unix versão 6 para a plataforma Intel (xv6, desenvolvido no MIT); **b)** analisar e planejar a implementação de dois escalonadores de processos; **c)** implementar os escalonadores; **d)** avaliar os escalonadores através de experimentos; **e)** elaborar relatórios descritivos para cada um dos trabalhos.

**Plataforma:** xv6 → <http://pdos.csail.mit.edu/6.828/2012/xv6.html>

**Modalidade:** individual

**Descrição do Trabalho I:** Implementar o escalonador de processos baseado em loteria (*lottery scheduling*). Na instanciação de um processo, deve-se passar ao sistema a quantidade de bilhetes que o novo processo recebe. Caso o usuário não forneça esse dado, o sistema assume um número *default* de bilhetes. Assumir também um número máximo de bilhetes que um processo pode receber.

**Descrição do Trabalho II:** implementar o escalonador de processos *stride scheduling* (escalonamento em passos largos). Semelhante ao escalonamento por loteria, cada processo recebe um número fixo de bilhetes (*tickets*). Ao invés de utilizar um sorteio (abordagem probabilística), calcula-se o “passo” (*stride*) de cada processo como sendo o resultado da divisão de um valor constante (e.g., 10.000) pelo número de bilhetes do processo. Cada processo inicia com uma “passada inicial” igual a zero (0). O escalonador seleciona o processo com o menor valor de passada atual; portanto, inicialmente qualquer um dos processos prontos pode ser selecionado. Nesse caso, utiliza-se um critério qualquer para desempate (e.g., o processo com o maior PID é selecionado). Após selecionado, a passada do processo é incrementada com o valor do “passo” do processo. Por exemplo, assuma que há 3 processos, A, B e C, cada um com 100, 50 e 250 bilhetes, respectivamente; portanto, A tem passo 100, B tem passo 200 e C tem passo 40. Caso o processo A seja selecionado inicialmente, a sua passada é atualizada para 100. Em seguida B e C ainda tem passada zero. Vamos assumir que B seja escalonado, atualizando sua passada para 200. Em seguida, o processo necessariamente selecionado é o C (passada ainda é zero), fazendo sua passada ser atualizada para 40. Em sequência, o processo C é selecionado mais duas vezes seguidas até atingir uma passada maior que o processo A (passada = 100). Essa abordagem torna o escalonamento **determinístico**, concedendo rapidamente uma fração da CPU correspondente ao número de bilhetes de cada processo.

**O que entregar (T1 e T2):** Além do código completo, incluir um relatório (em formato de artigo no modelo da SBC em latex) contemplando: **(a)** descrição do planejamento e implementação do escalonador: **o que precisou ser modificado/acrescentado(s) no xv6 (sempre citando os fragmentos do código fonte correspondentes à explicação/descrição apresentada); (b)** estratégia e aplicativos desenvolvidos para avaliar o funcionamento/desempenho do escalonador; **(c) Conclusões.**

#### Datas importantes:

- **16/10/2017:** Entrega (via *moodle*, se possível<sup>1</sup>) do Trabalho 1 em um arquivo comprimido contendo **todos os fontes do xv6** e o relatório no formato PDF (nomeado **T1.pdf**);
- **30/11/2017:** Entrega (via *moodle*, se possível<sup>2</sup>) do Trabalho 2 em um arquivo comprimido contendo **todos os fontes do xv6** e o relatório no formato PDF (nomeado **T2.pdf**);
- **Datas das apresentações: a serem definidas (após o prazo final de entrega de cada trabalho).**

1 Caso o tamanho do arquivo exceda o limite estabelecido no *moodle*, armazene o arquivo na rede e descreva como proceder para baixá-lo (submeta um arquivo “T1.txt” com as instruções necessárias).

2 Vide nota anterior.