

## Trabalho Prático 1 Escalonamento de Processos

Dado um simulador de carga de trabalho para o subsistema de gerenciamento de processos de um sistema operacional hipotético, implemente o algoritmo Lottery Scheduling como um dos algoritmos suportados pelo simulador. Seu Lottery Scheduling deve ser implementado com número indefinido de tickets e com a possibilidade de atribuição de múltiplos tickets a cada processo. Deve ser suportada a troca de tickets entre processos.

O simulador encontra-se escrito em linguagem C e é composto pelos seguintes arquivos:

process.h – Arquivo de cabeçalho para a declaração da API de manutenção de processos  
process.c – Arquivo que implementa a API de manutenção de processos  
sched.h – Arquivo de cabeçalho para a declaração da API de escalonamento de processos  
sched.c – Arquivo com a implementação do escalonamento de processos  
*lottery.h – Arquivo de cabeçalho para a declaração da API baseada no Lottery Scheduling*  
*lottery.c – Arquivo com a implementação do escalonamento baseado em Lottery Scheduling*  
main.c – Arquivo do programa principal do simulador

Todo o seu desenvolvimento deve estar contido nos arquivos lottery.h e lottery.c, destacados acima. Você somente pode alterar esses arquivos e fazer uso apenas das APIs disponibilizadas nos arquivos de cabeçalho, além é claro, da API padrão C.

Toda a descrição das API's está contida nos respectivos arquivos de cabeçalho.

As estruturas e funções a serem implementadas possuem explicação no arquivo de cabeçalho lottery.h. São elas:

```
typedef struct lottery_params LotterySchedParams;  
  
void lottInitSchedInfo(void);  
  
void lottInitSchedParams(Process *p, void *params);  
  
void lottNotifyProcStatusChange(Process* p);  
  
Process* lottSchedule(Process *plist);  
  
int lottReleaseParams(Process *p);  
  
int lottTransferTickets(Process *src, Process *dst, int tickets);
```

Quaisquer funções auxiliares não podem ser externalizadas no arquivo de cabeçalho lottery.h. Identificadores, tipos, assinaturas de função, enfim, linhas já escritas de programa, não podem ser modificadas.

Quaisquer funções necessárias e não implementadas na API de processo ou na API de escalonador devem ser discutidas e solicitadas ao professor. O mesmo vale para possíveis erros detectados.

O atendimento se dará somente de forma agendada e fora do horário de aula.