



Programação Paralela - Python

anthony@computacao.cua.ufmt.br



Forking

- Forking é uma ferramenta exclusiva de sistemas de bases unix
- Consiste em criar uma cópia do programa e executar essa cópia em paralelo com a original



Forking

- Note que nos exemplo anteriores os nossos programas simplesmente chamam uma função do programa original
- Podemos também montar programas que são completamente independentes, ao utilizar a chamada da função `os.exec__`



Forking

- `os.execv(programa, sequenciaLinhaComando)`
 - O programa cujo nome é passado é executado com os argumentos passados, que pode ser uma lista ou tupla
- `os.execl(programa, cmdarg1, cmdarg2, ..., cmdargn)`
 - É passado o nome do programa a ser executado com cada um dos argumentos da linha de comando passados como argumentos independentes da função
- `os.execlp`
 - python irá localizar o diretório do executável usando o sistema de procura de caminhos
- `os.execlp`
 - permite um último argumento que consiste de variáveis do sistema a serem mandadas ao programa
- `os.execvpe`
 - Combina as funcionalidades dos dois últimos



Threads

- ▀ `_thread`
- ▀ As threads serão como nossos forks
- ▀ Mas executaremos objetos em um mesmo processo
- ▀ Garantindo melhor
 - ▀ Performance
 - ▀ Simplicidade
 - ▀ Compartilhamento de memória
 - ▀ Portabilidade



Threads

- Problemas mais notáveis de threads é para sincronizar operações
- Devemos usar estrutura de dados para gerenciar as regiões críticas
 - Mutex permite um acesso organizado a determinado recurso do programa de maneira simples
- Consigo criar pontos dentro do código para verificar quais threads estão executando e quais já terminaram



Threading

- Implementa objetos baseados no módulo `_thread`
- Performance muito superior a `_thread`



Exercício

- O arquivo txt possui múltiplas linhas contendo números. Escreva um objeto thread que consiga acessar uma linha específica do arquivo e lê-lo até certo ponto de forma a obter a soma dos números contidos dentro do arquivo
- Agora, construa vários arquivos txt contendo números inteiros, e cada thread fica responsável por abrir pelo menos 1 e fazer a somatória de todos os números. Ao final, retorne o somatório geral de todos os arquivos .txt



Queue

- Módulo que organiza o acesso a um tipo de dado qualquer
- Modelo segundo o conceito FIFO
- São semelhantes a listas
- Mas são controladas por uma thread lock



Comunicação Inter processual

- ▶ Pipes nos permite criar caminhos entre um arquivo e outro e mandar dados de um arquivo para o outro
- ▶ Podemos usá-los para mandar dados entre threads
- ▶ Há 2 tipos de pipes
 - ▶ Anônimos
 - ▶ É criado instantaneamente
 - ▶ Nomeados
 - ▶ Baseado em um arquivo pré existente ou recentemente criado