

Memória Virtual

Implementar um programa simula a falta de páginas (*page fault*) de processos em um sistema operacional.

1. O programa deve funcionar em dois modos: sequencial e aleatório
2. No modo sequencial ele lê a criação de processos, término de processos, alocação de memória e acessos seguindo uma lista de comandos conforme exemplo abaixo.
3. No modo aleatório o programa deve:
 - a. O programa deve criar um conjunto de threads para simular processos executando.
 - b. Cada processo possui um tamanho, que representa quantos bytes ele ocupa na memória.
 - c. O processo passa o tempo todo: solicitando acessos endereços aleatórios de memória.
 - d. O processo pode também alocar mais memória com uma probabilidade de I%.
 - e. O processo termina com uma probabilidade de J%.
4. O gerente de memória deve alocar o número de páginas para o processo, relativo ao tamanho do processo.
5. Para cada acesso, é necessário verificar se a página do processo onde aquele endereço se encontra, está ou não presente na memória. Se estiver o “acesso” é realizado sem problemas. Se não estiver, então o gerente de memória deve ser acionado e um “dump” da memória deve ser realizado, as tabelas de páginas dos processos, a situação da memória (que processo está ocupando cada página), e o endereço que gerou o *page fault*.
6. Deve haver alguma forma de acompanhar (visualizar) o que está acontecendo no programa a cada solicitação ou liberação.

As informações de maneira manual (por arquivo) possui o seguinte formato:

Exemplo:

```
Modo: manual ou aleatório
lru/aleatório/....
Tamanho da página
Tamanho da memória física (múltiplo do tamanho das páginas)
Tamanho da área para armazenamento das páginas em disco
C|A|M|T processo [tamanho|endereço|]
```

Onde:

- C X Y – cria um processo X com tamanho de memória Y
- A X Z – processo X acessa endereço de memória Z
- M X W – processo X aloca mais W endereços de memória
- T X – processo X termina
- X, Y, Z, e W são números naturais

Entregar código e documentação, conforme formato fornecido.