

Projeto de SO

A aplicação de memória nesta etapa do projeto será dada da seguinte forma. Até o momento os processos (Requisições à calculadora) são enviados de forma concorrente ao recurso compartilhado (Calculadora). Desta forma, adicionamos um bloco de memória que pode ser dividido contiguamente. Consideramos agora que os processos gastam memória. Iremos alocar processos à memória com o algoritmo First-fit. Desta forma nem todos os processos serão executados rapidamente, pois deverão entrar em estado de espera.

A memória das equações do processo será calculada da seguinte forma: Subtração ou adição representam 1 unidade, divisão ou multiplicação representam 3 unidades e exponenciação conta 5 unidades.

Ex:

$1\ 3\ 2\ 7\ 4\ -\ +\ -\ +$	$= (2 + e\ 2\ -)$	= 4 de memória
$1\ 2 + 3\ 4\ -\ +$	$= (2 + e\ 1\ -)$	= 3 de memória
$1\ 2 + 3 * 2 /$	$= (1 +, 1 / e\ 1 *)$	= 7 de memória
$2\ 4 * 6\ 2 ^\wedge -$	$= (1 -, 1 * e\ 1 ^\wedge)$	= 9 de memória
$2\ 3 * 2 ^\wedge$	$= (1 * e\ 1 ^\wedge)$	= 8 de memória

Sorteamos a quantidade de memória entre 50 unidades de memória e executamos o algoritmo como explicado acima. Para simplificar a implementação não utilizaremos concorrência, mas simulamos isto colocando de forma aleatória todos os processos num array. Iremos enviar processo a processo na ordem do array para a calculadora. Alocamos os processos na memória até que ela esteja cheia, e os executamos na ordem que foram alocados. A cada processo que saia da memória checamos se há espaço disponível para o próximo ser alocado. Encerramos o algoritmo quando não houver mais processos no array ou caso o tamanho de um processo seja maior do que a memória.

Ex: No caso abaixo seguindo o passo a passo acima colocaremos a memória da seguinte forma. O processo P2 não será alocado pois é muito grande para a memória

