PROJETO 3º BIMESTRE
SISTEMAS COMPUTACIONAIS

Como vimos em aula todos os programas que concorrem para o uso do processador devem primeiramente estar carregados na memória primária de um sistema computacional, após o carregamento estes programas entram na final de processos aptos para serem processados. Podemos ver na figura abaixo uma típica representação das etapas para execução de um processo.

A figura acima trata justamente da execução de um único processo no processador, mas em um ambiente computacional atual o que temos é a execução de diversos processos, portanto este modelo de referencia deve ser adaptado para a confecção do projeto.

O projeto consiste na simulação do carregamento de um programa na memória, passagem na fila de aptos e a execução deste no processador.

Vamos supor que 10 blocos de memória permitem o carregamento de 10 programas e isso pode ser representado facilmente por meio de um vetor de 10 posições na linguagem **C**.

Também iremos simular a fila de aptos e suspensos por meio de vetores de 10 posições.

Caso o programa apto ganhe o direito de uso do processador ele pode ser executado até o final, ser suspenso ou ser colocado novamente na fila de aptos (situação que ocorre quando o tempo de uso do processador foi excedido – *tempo de quantum*). Estes 3 estados devem ocorrer aleatoriamente na simulação.

1. Executar até o final.
2. Fim do tempo de quantum
3. Suspensão (Espera de um dano disco ou dispositivo de IO- entrada e saída)

O programa deve contar com um menu de opções este menu deve conter os seguintes itens

1. Carregar o programa na memória
    a. Não permitir que mais de 10 programas ocupem a memória ao mesmo tempo (seu vetor que simula a memória possui apenas 10 posições).
    b. Permitir que o usuário digite **uma letra** para representar um processo ou programa que será carregado na memória.
    c. Não permitir que existam processos com a mesma letra de identificação.

2. Listar programas na memória (mostrar todos os valores do vetor que simula a memória).
3. Listar programas na fila de aptos (mostrar todos os valores do vetor que simula a fila de aptos).
4. Listar programas na fila de suspensos (mostrar todos os valores do vetor que simula a fila de suspensos)
5. Enviar programa da memória para fila de aptos (tenha como a entrada de dados a letra que representa o processo)
    a. Caso a letra não corresponda a nenhum processo envie uma mensagem para o usuário do sistema.
6. Executar programa (Ciclo do processador)
    a. Nesta opção o programa deve verificar se existe algum processo na fila de aptos. Caso existe a posição zero da lista é enviada para o processador.
    b. Após a simulação do processo de um programa o sistema deve remover da posição 0 do vetor de aptos o processo que acabou de ser executado e mover o processo da posição 1 do vetor para a posição 0 e a posição 2 do vetor para a posição 1 e assim por diante. Caso o processo não tenha terminado posicione-o na ultima posição do vetor de aptos.
    c. Caso o processo tenha sido terminado remova-o da memória e adicione-o no vetor de programas terminados.
7. Tirar programa da fila de suspensos (tenha como a entrada de dados a letra do processo)

Tendo como base as especificações acima teremos 4 vetores.

O vetor 1 representa a lista de programas carregados na memória, cada letra representa um programa diferente, as letras são definidas por meio da digitação do usuário:

Vetor 1

p	w	r	d	s

Lista de programas aptos para serem executados. No vetor abaixo apenas os processos **d** e **r** estão aptos para serem processados.

Vetor 2

d	r			

Lista de processos suspensos (que esperam por dispositivos de entrada e saída). No vetor abaixo o processo s espera por um dado no HD.

Vetor 3

s				

O vetor de programas terminados deve possuir 512 posições, abaixo podemos ver que os processos que já foram executados e terminados

Vetor 4

x	z	k	y	h

Código necessário para sorteador números.
Utilize este código para simular os estados de um processo:

0. Executa até o final.
1. Fim do tempo de quantum
2. Suspensão (Espera de um dano disco ou dispositivo de IO- entrada e saída)