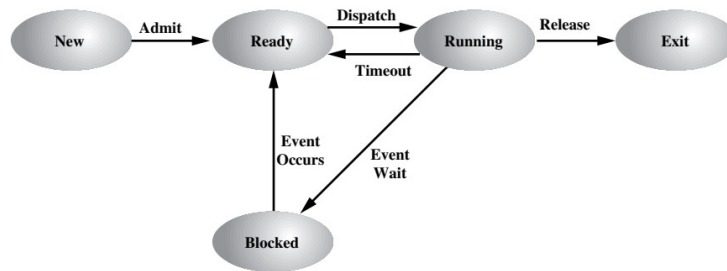
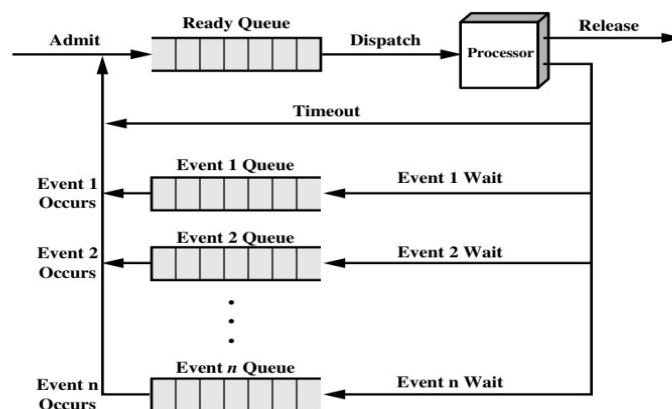


Simulador de Gerência de Processos

O trabalho tem como objetivo a implementação de um simulador para o gerenciamento de processos seguindo o modelo de 5 estados, conforme ilustra a figura abaixo:



Assuma que o SO usa um sistema de múltiplas filas para solicitações aos diversos dispositivos de entrada e saída, conforme exhibe a figura abaixo:



Implemente um programa em C (ambiente Linux) que tem como entrada:

- a descrição do ambiente: consiste no nº de dispositivos não concorrentes e no tempo de serviço de cada dispositivo;
- a descrição de vários processos: composta pelo tempo de chegada, tempo de uso da CPU e tempos de execução nos dispositivos, por processo.

Cada processo alterna períodos de uso do CPU com solicitações de uso de dispositivos. O programa deve gerar uma saída com o trace dos processos apresentando, para cada processo, em cada instante, o seu estado. Ao final, o programa deve exibir, para cada processo, o tempo total de espera e o tempo total de uso da CPU.

Formato de input:

```
<qtde_proc> <qtde_disp> <t d1> <t d2> ... <t dn>
<início P1> <t_CPU> <disp> ... <t_CPU> <disp>
...
<início Pn> <t_CPU> <disp> ... <t_CPU> <disp>
```

Exemplo de input:

```
3 3 2 5 7
0 5 1 1 3 1
2 2 2 3 3 5 2 4
3 7 2 2
```

Formato de output:

```
<0> <estado de P1> ... <estado de Pn>
...
<t> <estado de P1> ... <estado de Pn>
...
<t_final> <estado de P1> ... <estado de Pn>
-----
<tempo de espera de P1> <tempo de CPU de P1>
...
<tempo de espera de Pn> <tempo de CPU de Pn>
```

Observe os seguintes detalhes de implementação:

- A entrada (input) e a saída (output) devem possuir os formatos descritos anteriormente.
- A entrada (input) deve ser lida de um arquivo texto (.txt) e a saída (output) deve ser salva em um arquivo texto (.txt).
- O tempo avança em unidades discretas (... , 4, 5, 6, ...).
- O último ciclo de cada processo deve ser um ciclo de CPU.
- Os ciclos de CPU devem ser maiores que zero.
- O instante inicial de um processo P_i é o instante em que o processo entra no estado "new".
- O programa deve criar um "PCB" para o processo e adicioná-lo em uma fila de processos no estado "new".
- Todas as filas estão inicialmente vazias e devem funcionar em modo FIFO (First-In-First-Out).
- A admissão de processos da fila "new" para a fila "ready" apenas deve ocorrer se o número de processos na fila "ready" for menor ou igual a 2.
- A transição para o estado "ready" se dá verificando os processos que se encontram nas filas de dispositivos, na fila "new" e na CPU (nesta ordem).
- A verificação das solicitações de I/O é feita por ordem crescente do número do dispositivo.
- Cada dispositivo somente começa a processar uma solicitação após as anteriores terem sido satisfeitas.

- Os tempos de uso da CPU e de dispositivos devem ser valores inteiros.
- Um processo sai do estado "running" (CPU) se acabar sua execução ou se expirar sua fatia, que é de 4 unidades.
- Quando um processo termina, ele é inserido na lista do estado "terminated".
- A simulação deve terminar quando todos os processos se encontrarem no estado "terminated".

Orientações e informações importantes:

- Trabalho individual.
- A interpretação da especificação do trabalho faz parte da avaliação.
- O código deve estar organizado, indentado e obedecer criteriosamente à especificação.
- Não devem ser inseridos comentários no código.
- O programa deve estar modularizado (funções, com escopo definido) e utilizar passagem de parâmetros, evitando o uso de variáveis globais sempre que possível.
 - Também deve ser evitados os trechos redundantes de código, os quais podem ser resolvidos com o auxílio de funções.
- Certifique-se de compilar e testar exaustivamente o programa antes do envio (compilar com a flag -Wall).
 - Devem ser entregues:
 - arquivo .c, denominado **t3-simulador.c**, contendo a implementação da solução;
 - 2 casos de teste: arquivos .txt de input e seus respectivos outputs, os quais devem simular 2 execuções distintas com, no mínimo, 5 processos cada;
 - arquivo .pdf contendo um relatório descrevendo (máx. 3 páginas):
 - a implementação das filas;
 - a estrutura de dados que implementa o processo (PCB);
 - a estrutura do simulador, explicando os procedimentos executados em cada ciclo de execução.
 - os arquivos .c, .txt (4 arquivos, sendo 2 de input e 2 de output) e .pdf devem ser organizados em um arquivo .zip.
 - A entrega do trabalho deve ser feita, obrigatoriamente, pelo Moodle. Não serão aceitos trabalhos enviados fora da plataforma Moodle.
- A apresentação do trabalho será agendada posteriormente. Trabalhos não apresentados serão desconsiderados.
- Não serão analisados códigos .c antes do período de entrega.
- Não serão aceitos trabalhos fora do prazo, nem mesmo com desconto na nota.
- Trabalhos total ou parcialmente copiados receberão nota zero.