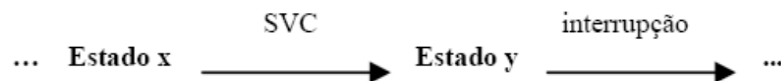


- 1) (3,0) A partir do diagrama completo de transição de estados para processos em UNIX, apresente uma possível sequência de estados referente ao seguinte histórico de um processo: “O processo foi criado e iniciou a execução de instruções comuns (CPU em modo usuário), sem deixar a CPU até a ocorrência de uma requisição de E/S, a qual demanda um tempo “longo” para ser atendida. Durante esse tempo, o processo sofreu a ação do escalonador de médio prazo (swapper). Uma vez atendida a requisição de E/S, o processo voltou a executar instruções comuns, até ser preemptado. Ao voltar para a CPU, o processo prosseguiu executando instruções comuns até o seu término”.

Nota1: Estados do diagrama completo UNIX são: *Initial*, *Ready* (pronto), *Stopped* (pronto suspenso), *Asleep* (bloqueado), *Stopped+asleep* (bloqueado suspenso), *Kernel Running*, *User Running*, *Zombie*.

Nota 2: É necessário associar as transições presentes na sequência de estados a cada evento listado no histórico. Um exemplo fictício de resposta é mostrado abaixo :



- 2) (2,5) Por que os algoritmos de escalonamento SJF (*Shortest Job First*) e o SRTF (*Shortest Remaining Time First*) são tão difíceis de ser implementados? Descreva uma alternativa (algoritmo ou política) de escalonamento que possa ser implementada mais facilmente e que consiga atingir os mesmos objetivos do SJF.
- 3) (2,0) No UNIX, um processo pode encontrar-se no estado *Kernel Running* (CPU em *Kernel Mode*) enquanto o sistema pode apresentar dois contextos de execução, *Process Context* e *System Context*. Explique a diferença entre esses dois contextos de execução.

- 4) (2,5) Um grafo de precedência é um grafo direcionado em que a relação (a) → (b) indica que 'a' precede 'b'. Na figura ao lado, os nós do grafo representam funções que devem ser executadas por processos. Dessa forma, o grafo de precedência ilustrado representara a ordem em que as funções serão executadas. Usando as chamadas `fork()` e `waitpid()` no UNIX, implemente um programa em C que seja capaz de garantir a relação de precedência e o paralelismo na execução das funções “Sx()” conforme indicado no grafo.

