

Tipos de sistemas operacionais.**1 - Introdução**

Quanto aos tipos, os sistemas operacionais podem ser:

- Sistema monoprogramável: batch;
- Sistema multiprogramável: batch, de tempo compartilhado e de tempo real.

2 - Sistema monoprogramável

O sistema monoprogramável é caracterizado pela execução de uma única tarefa [job, processo] de cada vez. Todos os recursos computacionais [processador, memória, periféricos] são dedicados à execução dessa tarefa.

A característica desse sistema é a subutilização dos recursos de processamento, memória e periféricos, pois somente é capaz de atender um de cada vez.

Por exemplo, enquanto o sistema estiver processando determinado job, eventualmente pode

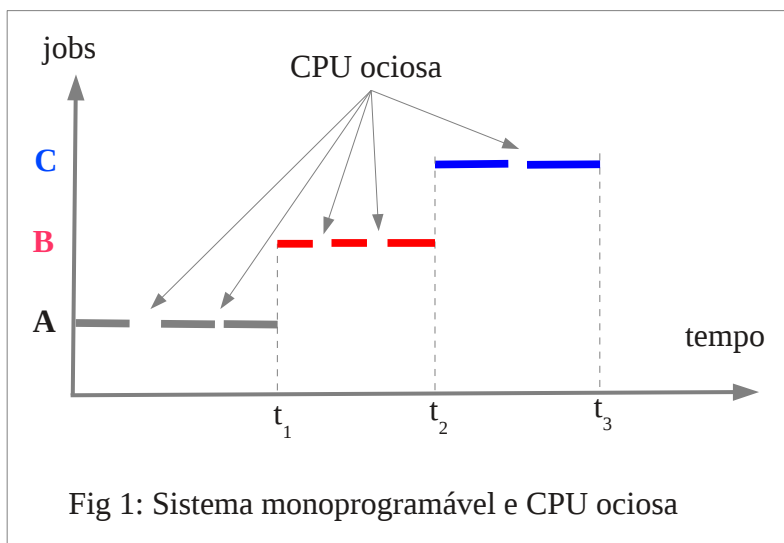


Fig 1: Sistema monoprogramável e CPU ociosa

parar esse processamento para aguardar algum evento qualquer [exemplo: I/O], e nesse período o processador ficar ocioso. Enquanto isso, existe uma fila de processos aguardando esse job finalizar.

A figura 1, ao lado, mostra um sistema monoprogramável com 3 jobs, A, B e C. Em t_1 , termina a execução do job A e inicia o job B, em t_2 finaliza o job B e inicia o job C. Durante a execução desses jobs algum evento externo levou a um estado de espera pela finalização

desse evento, enquanto isso o processador ficou ocioso.

3 - Sistema multiprogramável

Os sistemas multiprogramáveis permitem que várias tarefas [processos, jobs] residam na memória e concorram pelo uso dos recursos disponíveis. Entretanto, em qualquer momento apenas um processo ou job detém o uso do processador. Esses sistemas são mais complexos e eficientes que os monoprogramáveis.

A multiprogramação permite que o processador atenda a outra tarefa em condições de executar, caso a tarefa que estava em execução entre num estado de espera por evento externo [por exemplo, ler dados numa fita].

O sistema precisa então gerenciar os recursos computacionais de forma ordenada e protegida, para atender às tarefas concorrentes. Concorrentes no sentido de que os diversos processos disputam entre si pelo uso desses recursos.

No exemplo da figura 2 ao lado, enquanto a tarefa A aguarda um evento externo, a CPU processa a tarefa B, e enquanto a tarefa B aguarda um evento externo, a tarefa C é processada. Notar que ao final as tarefas terminam num tempo t_3 menor do que num sistema monoprogramável.

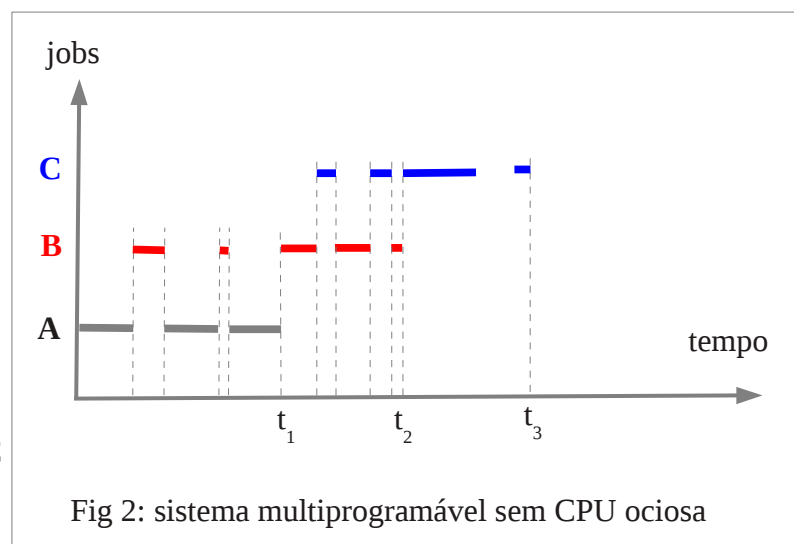


Fig 2: sistema multiprogramável sem CPU ociosa

O sistema multiprogramável, pela forma como as suas aplicações são gerenciadas, podem ser classificadas como do tipo batch [lote], de tempo compartilhado [timesharing] ou de tempo real. Um mesmo sistema multiprogramável pode suportar um ou mais desses tipos.

3.1 - Sistema batch

A principal característica do sistema batch é a ausência de interação com o usuário. A fila de jobs vai sendo processada à medida que haja disponibilidade de recursos para isso.

Modernamente, os sistemas operacionais implementam ou simulam o processamento batch, pois não existem mais sistemas dedicados a esse tipo de processamento.

O sistema em *batch* dominou a computação desde a década de 1950 até o começo da década

de 1970 e tinha por característica codificar os comandos (instruções para o sistema) em cartões ou fita que eram então colocados em uma leitora para que o sistema operacional pudesse ler e executar sequencialmente.

Embora ultrapassado, esse modo de processamento em lote ainda é muito útil e existe hoje o conceito *batch*: um usuário escreve num arquivo de texto os comandos do sistema que quer executar, submete essa tarefa e então pode até se desconectar que o sistema irá seguir executando sequencialmente essa lista de tarefas. A utilidade prática então é permitir executar em *background*¹ tarefas repetitivas que não necessitem do usuário interativo. O arquivo de texto com as instruções é chamado de script, por exemplo *shell script*.

No batch moderno, o sistema operacional tem a liberdade de não precisar mais fornecer uma resposta rápida, poupando então recursos computacionais para os usuários interativos que não querem enfrentar a resposta lenta que ocorre quando o sistema está sobrecarregado. Isso é possível ao ser atribuído uma prioridade baixa ao script, que desse modo vai usar somente CPU ociosa.

Os sistemas Unix, Linux, Windows e Mac OS [entre outros] têm suporte a execução de programas em batch.

3.2 - Sistema de tempo compartilhado

O sistema de tempo compartilhado veio nos anos 1960 para atender aos usuários interativos, devido ao surgimento dos terminais. Inicialmente eram chamados de terminais burros, pois tudo que faziam era dar acesso aos recursos de uma máquina central, recursos esses que eram compartilhados simultaneamente por vários usuários.

Esse sistema concede uma fatia de tempo [time slice] de processador para cada terminal ativo [com usuário logado] num esquema de fila, de modo que cada terminal seja atendido um após o outro. Como a fila anda rápido, os usuários têm a sensação de que os recursos estão o tempo todo disponíveis para eles.

Esgotada a fatia de tempo de cada usuário, o processador é passado para o seguinte da fila. Modernamente, muitos sistemas podem alterar a prioridade do processo [tarefa, job] do usuário, alterando a sequência de atendimento ["furando essa fila"] ou modificando a duração dessa fatia de tempo.

Outro aspecto desse sistema é que memória, sistema de arquivos e periféricos também são compartilhados pelos usuários. No entanto, a sensação de qualquer usuário é de que todos os recursos computacionais estão lá ao dispor dele.

À exceção dos sistemas de tempo real, os sistemas operacionais modernos são de tempo compartilhado.

3.3 - Sistema de tempo real

¹ Background: um executável rodando em background não está preso ao terminal que foi usado para iniciar essa execução.

A principal característica do sistema de tempo real está no tempo de resposta exigido na execução das tarefas. Como exemplo, no sistema de tempo compartilhado o usuário interativo pode experimentar alguma espera [lentidão do sistema], já no de tempo real a resposta do sistema precisa estar dentro de limites rígidos que, se não cumpridos, inviabilizam o funcionamento.

Para garantir a resposta dentro de limites de tempo, o sistema permite que determinada tarefa [processo] permaneça usando o processador pelo tempo que for necessário, ou até que apareça outra atividade com maior prioridade ainda. Nesse sentido, não existe o conceito de tempo compartilhado.

O sistema de tempo real é necessário quando se exige respostas imediatas. Os usuários interativos preferem respostas rápidas, mas no caso do sistema de tempo real uma resposta muito rápida é necessária. Por exemplo, um sistema de controle de tráfego aéreo deve ser capaz de processar e manter a localização, altitude, velocidade e direção de todos os aviões nas cercanias de um aeroporto: qualquer atraso nessa resposta poderá ir além de um simples inconveniente e se tornar um sério acidente. Um outro exemplo clássico é o caso do direcionamento automático da trajetória de um míssil, que deve acionar comandos rapidamente e não depois que o míssil já destruiu o alvo errado.

4 - Sistema monotarefa e multitarefa

Modernamente, usam-se os termos monotarefa e multitarefa.

Por monotarefa entende-se um sistema monoprogramável e monousuário, como exemplo o antigo MS-DOS [1983/2001]. Nesse sistema, somente um usuário podia acessar de cada vez, e esse usuário só podia executar um processo de cada vez: antes de executar um comando precisava esperar o anterior finalizar.

Por multitarefa entende-se um sistema multiprogramável, e que pode ser monousuário ou multiusuário. A grande maioria dos sistemas modernos estão nessa categoria, por exemplo o Linux é multitarefa e multiusuário, enquanto que os sistemas operacionais de celulares normalmente são multitarefa e monousuários.

Num típico sistema multitarefa moderno rodam simultaneamente centenas de processos, que podem ser processos do sistema, de usuários ou mesmo processos de serviço [por exemplo, serviço de e-mail].