

# **Administração de Dispositivos**

Aula 13 – Dispositivos virtuais

# DISPOSITIVOS VIRTUAIS

- Dispositivos em geral, tais como leitoras, plotters, scanners, perfuradoras, impressoras, etc., trabalham mais **eficientemente** quando **solicitados em sua capacidade máxima**.
- Quando a relação entre as taxas de entrada/saída e de processamento é tal que o intervalo de tempo entre um uso e outro do mesmo periférico se torna não desprezível, a eficiência do sistema tende a ficar prejudicada.

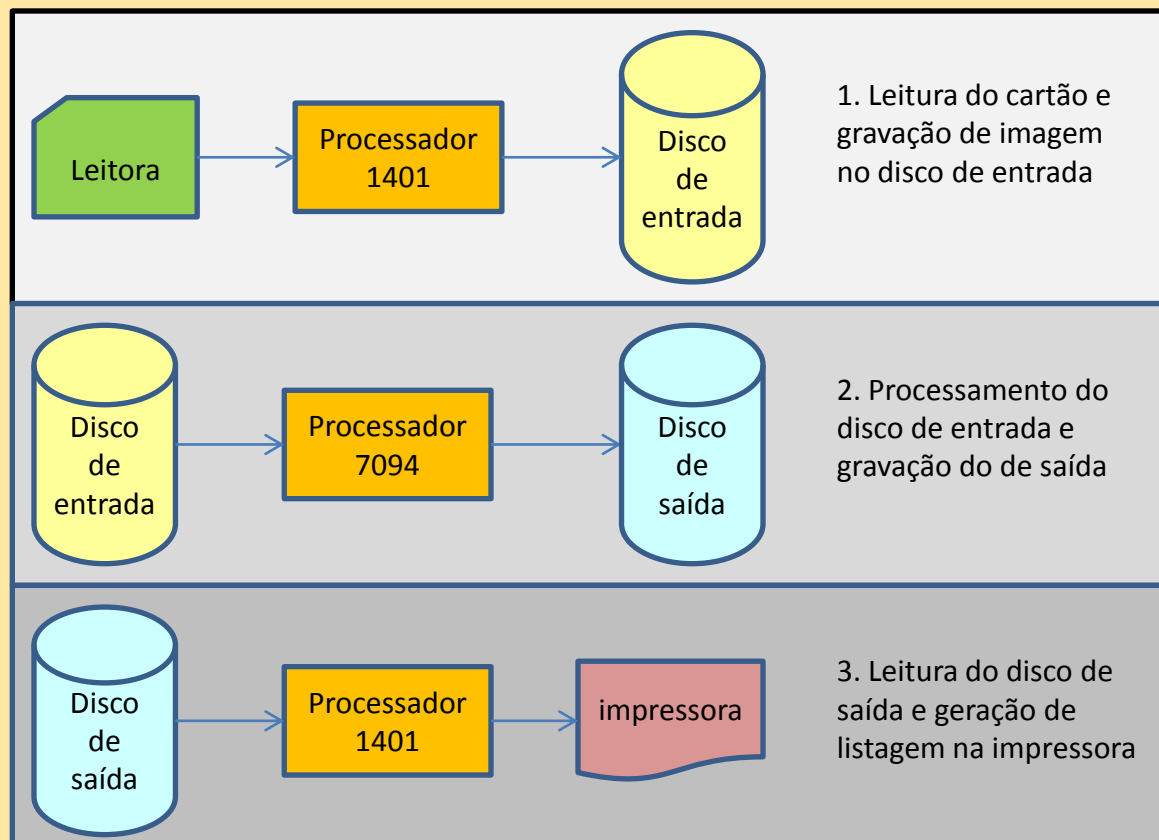
- Em determinadas situações, como ocorre nos casos de **alocação dedicada**, alguns dispositivos físicos podem, por sua natureza, permanecer alocados somente a um único job de cada vez.
- É o caso de scanners, impressoras, plotters, etc.
- Técnicas tais como as de "***Buffering***" e de **multiprogramação** tentam contribuir para reduzir o impacto destes problemas, atingindo parcialmente esta meta.

- "*Plotters*", "*displays*" gráficos, terminais seriais, etc., apresentam problemas parecidos.
- Um modo de resolver o problema, ou ao menos atenuá-lo consiste em **transferir toda a carga de entrada/saída dos dispositivos usuais para dispositivos de acesso rápido**, tais como tambores ou discos.
- Isto funciona relativamente bem, especialmente para dados bloqueados.
- Algumas das técnicas que têm sido utilizadas para isto são as seguintes:

# Operação "*off-line*" dos periféricos

- Um processador periférico incumbe-se de efetuar a leitura/impressão, usando memória de massa como meio intermediário.
- Efetua transferências de dados de um dispositivo para outro.
- Aparecem, nesta configuração, alguns problemas:
  - há necessidade de **intervenção humana** na operação do sistema
  - aumenta o "*turnaround time*", em consequência da ação do operador
  - há dificuldades nas políticas de "*scheduling*", ou ineficiência nas mesmas, devido ao fato de ser sempre conveniente completar o processamento de todo o conteúdo de um tambor ou disco antes de liberar para o usuário os resultados do processamento.

Uma configuração antiga que foi muito usada envolvia os computadores IBM 7094 (computador principal, trabalhando em multiprogramação) e IBM 1401 (auxiliar):



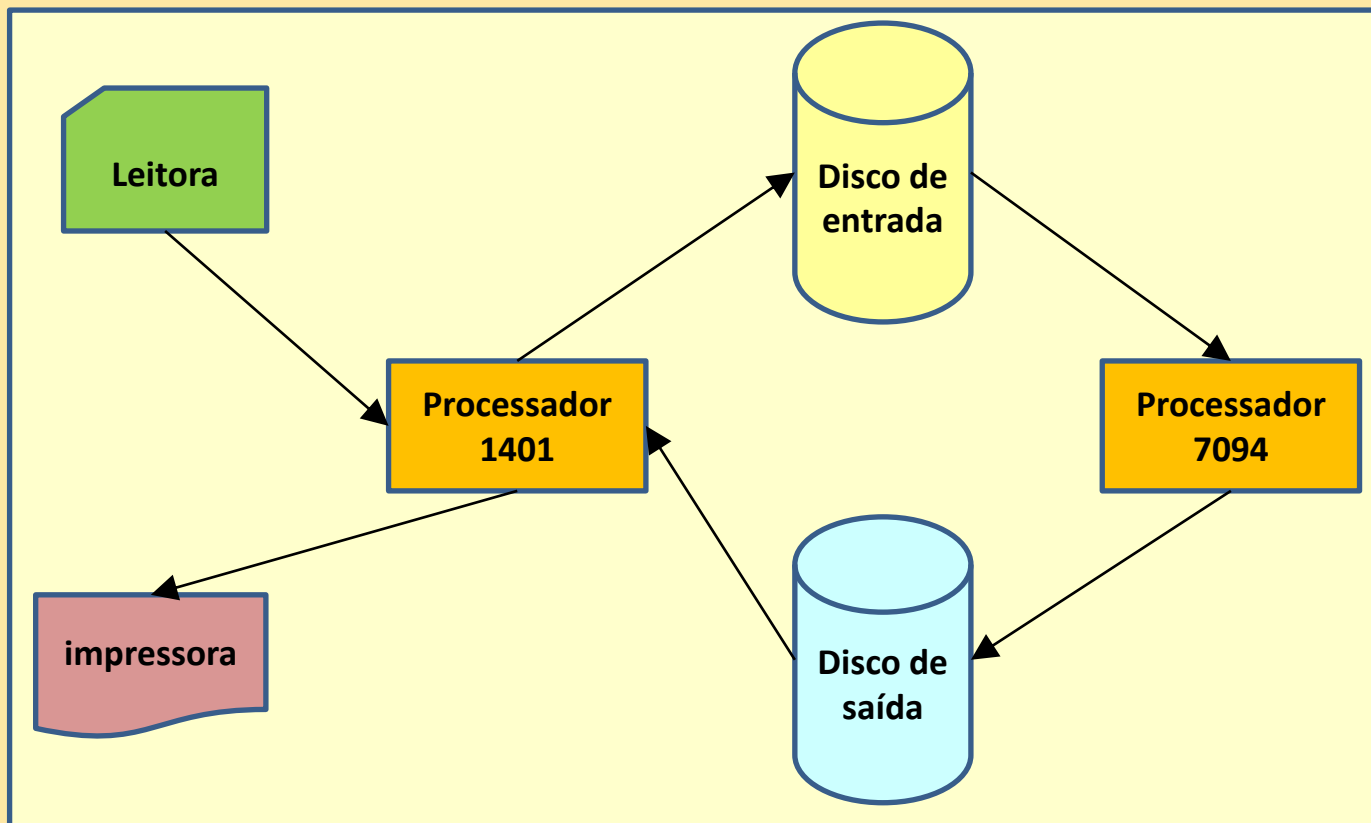
# Sistemas com acoplamento direto

- O principal problema do método "*off-line*" é o da necessidade de **movimentação física** dos discos de um sistema para outro, o que favorece a ocorrência de erros de operação.
- Para evitar tais inconvenientes, **conectam-se os dispositivos simultaneamente** ao computador principal e ao processador periférico.

- O mesmo par de processadores usado antes pode ser arranjado em configuração de acoplamento direto.
- Os sistemas 1401 e 7094 passam a competir, neste caso, pelo uso dos discos de entrada e de saída.
- **Não há mais intervenção manual** do operador para mover os discos de um sistema para o outro.
- Isso **exige um hardware** para decidir, em casos de conflito, qual dos dois sistemas deve utilizar o disco compartilhado.



Operação com **acoplamento direto**: os discos de entrada e de saída são compartilhados pelo processador principal (7094) e pelo processador periférico (1401), sem intervenção manual.



# Sistemas com Processador de Suporte associado

- Neste caso, **um processador adicional** é ligado diretamente ao processador central por meio de uma **via de alta velocidade**.
- O processador de suporte efetua toda a entrada/saída em disco, "***buffering***" e "***blocking***", e coordena **toda a entrada e saída** nos periféricos.

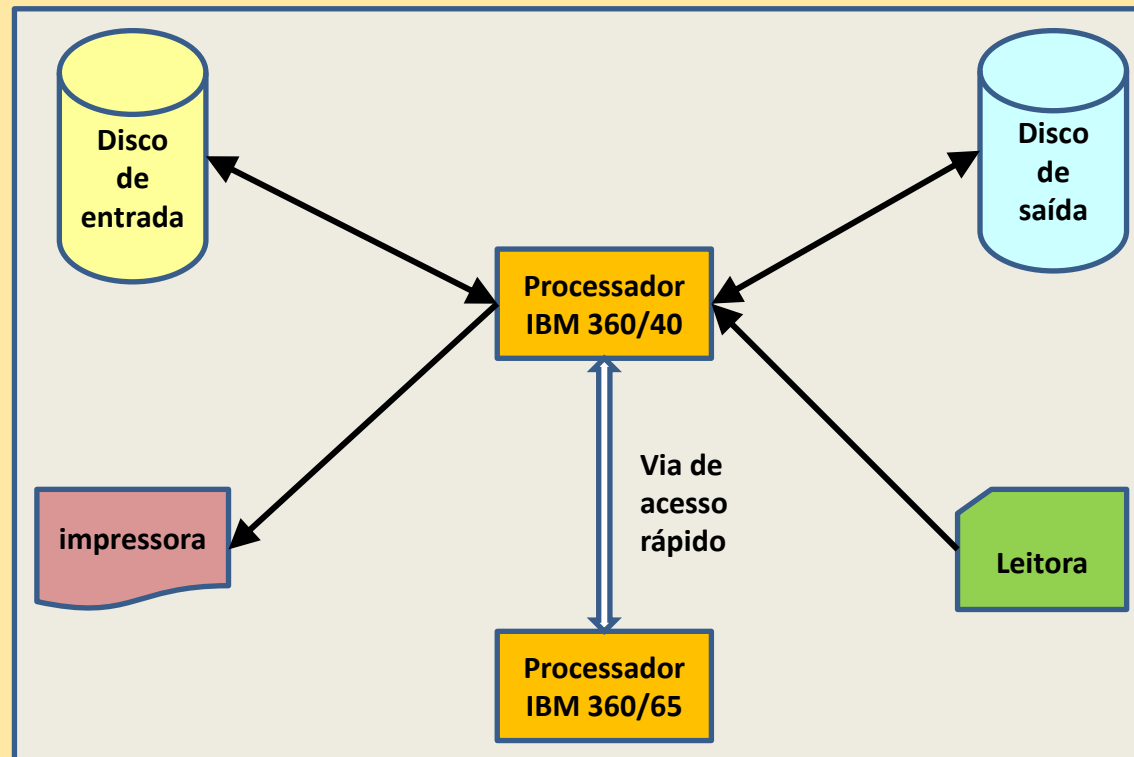
- Isso resolve os problemas de conflito que surgem pela utilização concorrente dos discos por parte dos processadores.
- Vantagem: pode simular **recursos inexistentes** (tais como, por exemplo, acesso aleatório e "*rewind*" em leitoras/perfuradoras de cartões ou em impressoras) o que permite ao sistema oferecer ao usuário recursos em número e qualidade diferentes dos fisicamente disponíveis.
- Desvantagem: se houver **sobrecarga** em um dos processadores, o outro pode ficar **ocioso**.

- Uma aplicação para essa configuração é utilizada pela IBM, envolvendo dois processadores/360: um modelo 40, funcionando como processador de suporte de um modelo 65, que é o processador principal.
- A figura seguinte esquematiza um sistema com processador de suporte associado.

## Sistema com processador de suporte associado:

O processador IBM 360/40 é encarregado de toda a entrada/saída.

O processador principal IBM 360/65 encarrega-se do processamento.

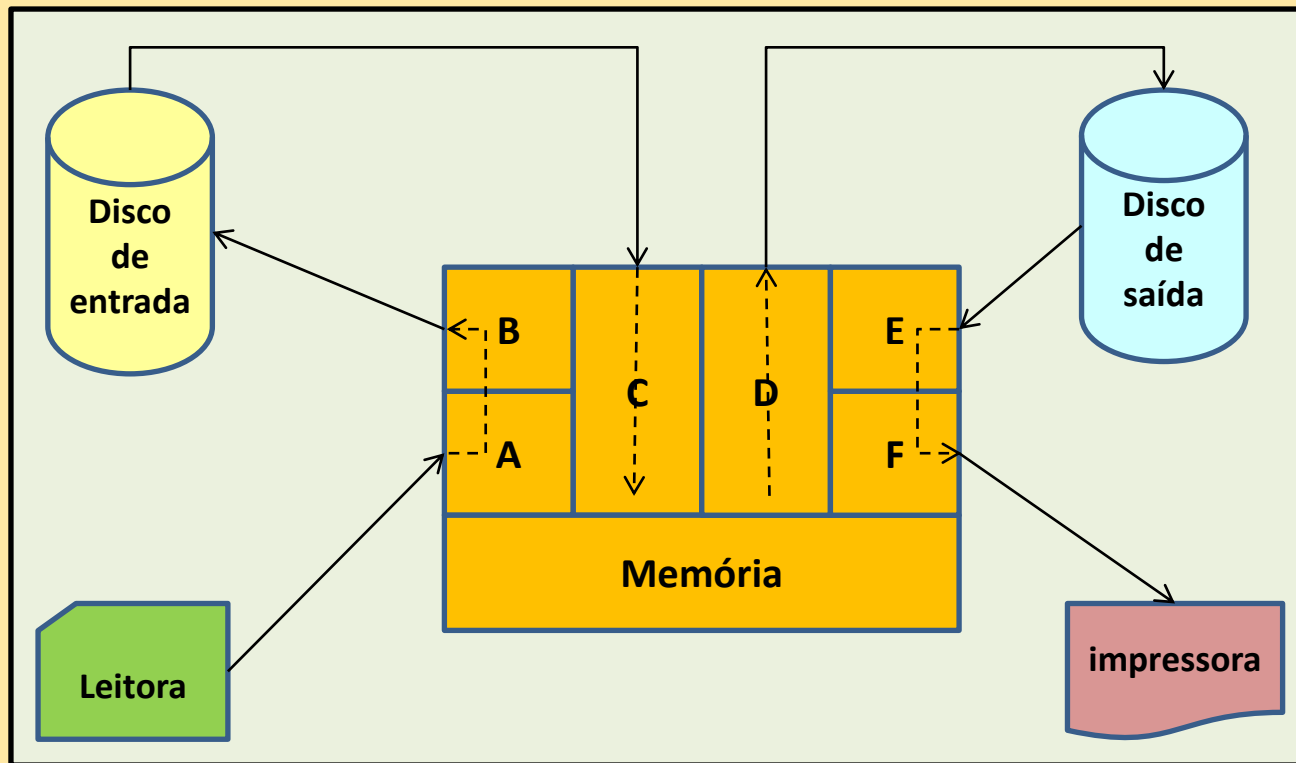


# Sistema virtual

- Os métodos apresentados utilizam processadores na tarefa de coordenação e de execução física das operações de entrada/saída.
- Uma boa alternativa para o uso de processadores para a execução de entrada/saída, e a utilização de **canais** para este fim (que, em última análise, também são processadores), evitando o uso de processadores adicionais no sistema.

- Com esta técnica, pode-se executar "***SPOOLing***" ("***Simultaneous Peripheral Operation On Line***"), que consiste basicamente na exploração dos canais para a execução física da entrada/saída.
- Isso permite ao sistema a **simulação do periférico**, dando ao usuário a possibilidade de efetuar entrada/saída em dispositivos simulados (virtuais), e tornando o sistema muito mais flexível.

# Esquema de um sistema virtual



Programas de canal:

- |                                   |                                   |
|-----------------------------------|-----------------------------------|
| (A) - entrada pela Leitora        | (B) - saída para disco de Entrada |
| (C) - entrada do disco de Entrada | (D) - saída para disco de Saída   |
| (E) - entrada do disco de Saída   | (F) - saída para a Impressora     |



- Na memória principal, os vários programas que estiverem sendo processados (em regime de multiprogramação) preenchem, na memória, as áreas de saída, e executam a leitura de áreas de entrada previamente preenchidos pelos dispositivos.

- A entrada de dados da Leitora para o buffer A se origina pela leitura física de cartões para a área A de entrada da Leitora, e é efetuada pelo programa de canal que trata essa Leitora.
- Esses dados são então transferidos para a área do buffer de entrada B do disco, de onde outro programa de canal promove sua transferência para o disco de entrada.
- Um terceiro programa de canal finalmente transfere os dados de entrada C do disco para a área de entrada do programa (Memória), de onde os dados são consumidos pelo programa em execução na memória principal.

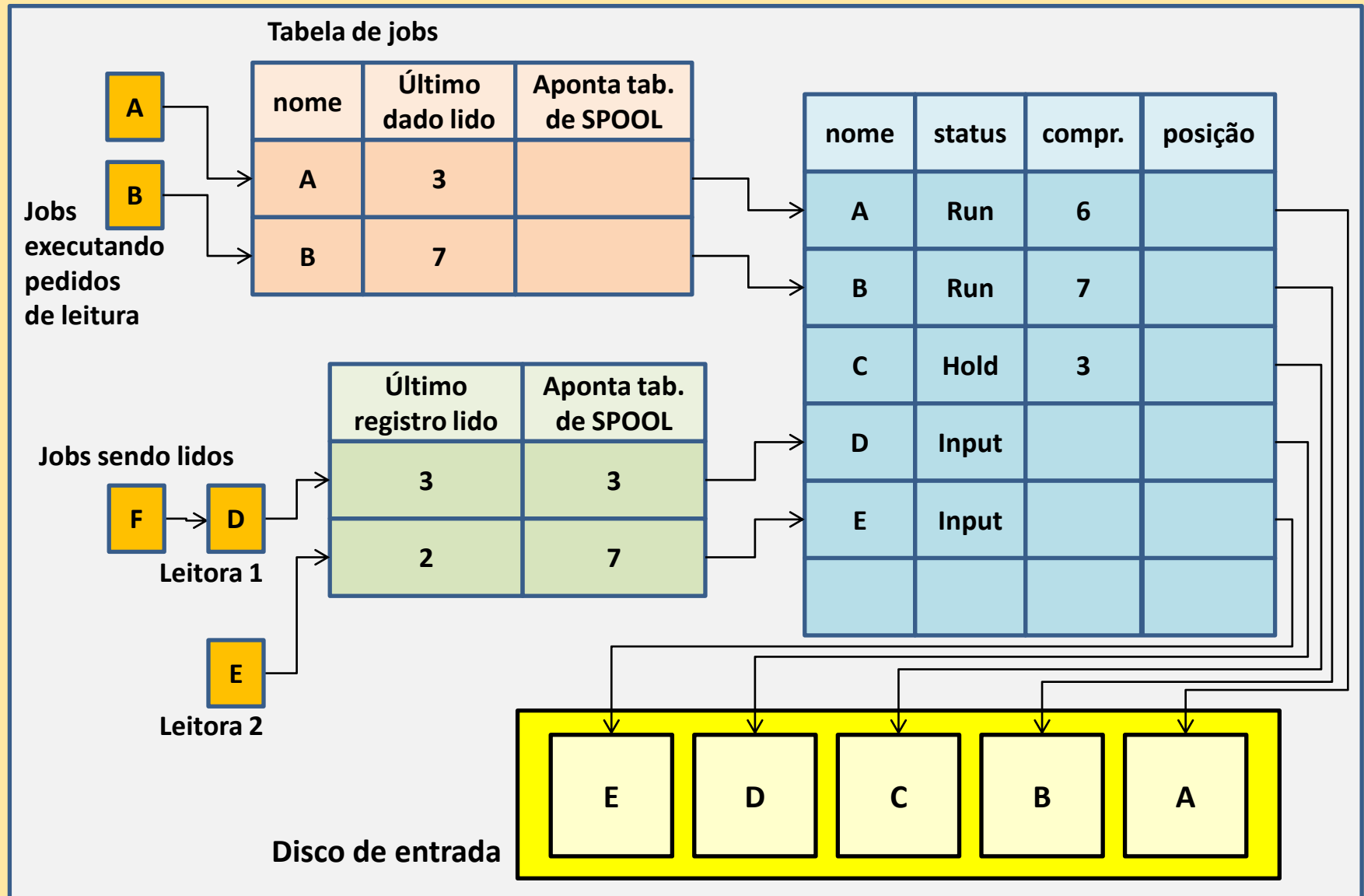
- A saída é análoga: dados originados na memória principal são depositados no buffer D de saída do disco, de onde um programa de canal os transfere para o disco de saída.
- Do disco de saída, os dados são copiados, através de um segundo programa de canal, para a área E de saída do disco, deixando-os disponíveis na memória, para serem impressos oportunamente.
- Um terceiro programa de canal completa a operação transferindo-os para a área F de saída, de onde são finalmente enviados para a impressão.

# Um sistema de "*SPOOLing*"

- Em muitos sistemas, os programas que implementam o "*SPOOLing*" são bastante complexos e interagem com os demais componentes do sistema.
- Em outros, eles fazem o papel de trechos inexistentes no sistema, executando algumas das funções que seriam desempenhadas pelo próprio sistema operacional.

- Neste exemplo, a rotina de "*SPOOLing*" é independente do restante do sistema.
- Ela incorpora basicamente duas chamadas de supervisor, uma para **leitura** e outra para **escrita**, que disparam as rotinas READNEXT e PRINTNEXT, respectivamente.
- A figura seguinte mostra uma estrutura de informação típica de um sistema de *SPOOLing*.

# Exemplo de *SPOOLing* (de entrada)



- A rotina de entrada faz o seguinte (a de saída é semelhante):
  - Faz a entrada dos dados a partir da leitora (antes do início do job)
  - Armazena-os tipicamente em um dispositivo de acesso direto
  - Busca nesse dispositivo uma cópia do dado quando for solicitada sua leitura pelo job (usa a rotina READNEXT do "*SPOOLing*").
  - Mantém na tabela de SPOOL de entrada a informação do estado do job :
    - "**RUN**" – os dados já foram lidos para o disco de entrada, e o job está em execução.
    - "**HOLD**" – os dados já foram lidos, mas o job aguarda para ser executado.
    - "**INPUT**" – a imagem do job ainda está sendo lida para o disco de entrada pelo sistema de "*SPOOLing*" .
  - O sistema de "*SPOOLing*" pode ler os parâmetros do job, ordená-los para facilitar o trabalho do "*Job scheduler*", e até trocar a ordem dos jobs se necessário, sem a intervenção do operador.