PCS 3446 – SISTEMAS OPERACIONAIS

Prof. João José Neto

AULA 15 - Administração de Informações

Função do Adm. de Informações

- A função de um administrador de informações no Sistema Operacional é análoga à de um almoxarife: deve cuidar do **armazenamento** das informações e do **atendimento de solicitações de acesso** a essa informação (localização e obtenção da informação).
- Ao conjunto de programas que executa tais funções no sistema operacional dá-se o nome de **Sistema de Arquivos** ("***file system***").

Funções do sistema de arquivos

- **Manter os dicionários** (*directories*) do sistema. Os dicionários correspondem aos itens do almoxarifado, e as informações ficam organizadas através de sua identificação (nome) e localização (endereço), no mínimo.
- Em um sistema operacional os V.T.O.C. (*volume table of contents*) contêm, além do nome e do endereço das informações contidas em um volume de armazenamento (rolo de fita, disco, *pen drive*, CD, DVD, etc.), algumas outras informações, das quais a mais importante é a que estabelece para os diversos usuários os respectivos direitos de acesso ao arquivo (p. ex. *Read only, Read/Write, Execute only*, etc.).

- Decidir a **política** de como e onde a informação deve ser armazenada
- Registrar quais usuários podem ter quais **tipos de acesso** à informação, visando, com isto:
  - Uso eficiente do meio de armazenamento secundário;
  - Acesso eficiente às informações;
  - Proporcionar flexibilidade para o usuário;
  - Proteger os direitos de acesso dos usuários às informações.

- **Alocar** o recurso (informação) ao processo solicitante, atualizando, quando for o caso, as informações de proteção, e permitindo ou não ao processo solicitante o acesso aos arquivos.

- **Recuperar** o recurso, liberando áreas temporárias, e atualizando convenientemente a informação, após o seu uso.

- Devem, conforme a conveniência caso a caso, permitir **informações temporárias e permanentes**
- Devem dispensar o usuário da tarefa de cuidar da formatação, da transferência física dos dados, das alocações de área em disco e do endereçamento físico da informação.
- Devem permitir ao usuário **compartilhar** e/ou **proteger** suas informações em relação aos outros usuários do sistema.

Terminologia

- É conveniente distinguir os seguintes termos, em gerenciamento de informações de um sistema operacional:
  - **Sistema de Arquivos** (***file system***) – trata da organização lógica simples das informações. **Não interpreta os dados**, nem é responsável pela estruturação física dos mesmos. Cada elemento que compõe um Sistema de Arquivos é denominado Arquivo (***file, data set***).
  - **Sistema de Gerenciamento de Dados** (***data management system***) – disponibiliza ao usuário ferramentas para a **estruturação física dos dados**. Para tanto, o sistema deve manter informações sobre os formatos físicos dos dados, apesar de não dispor de informações suficientes para interpretá-los.
  - **Banco de Dados** (***database system***) – oferece ao usuário ferramentas suficientes para que seja feito o acesso aos dados da maneira mais cômoda possível, o que exige que o sistema disponha de informações não apenas estruturais sobre os dados, mas também para permitir sua interpretação.

- Os gerenciadores de informação podem **variar muito em complexidade**, desde sistemas simples, destinados apenas a gerenciar arquivos do sistema operacional até os mais complexos, que permitem armazenar, acessar e compartilhar os dados de vários usuários, em modo interativo e em tempo real.

Um Sistema Simples de Arquivos

- Seja um comando de entrada/saída em arquivo, em alguma linguagem de programação típica de alto nível.
- Independentemente da linguagem em que for escrito, esse comando terá, provavelmente, um aspecto semelhante ao seguinte:

"LEIA O PRÓXIMO REGISTRO DO ARQUIVO (ARQ), QUE É O REGISTRO NÚMERO (R) E GUARDE SEU CONTEÚDO A PARTIR DA POSIÇÃO (X) DE MEMÓRIA".

- Supõe-se que o sistema disponha da informação do **comprimento do registro lógico** que se deseja ler.
- Os registros físicos contêm um **número de registros lógicos,** que também precisa ser conhecido pelo sistema.

- Em linguagens de alto nível, o **endereçamento** da informação é usualmente lógico, **simbólico**, ou seja, os arquivos são **referenciados por nome**, e não explicitamente por sua localização física.
- **Cada volume** de armazenamento (um disco, uma fita magnética, um DVD, um CD, um pen-drive), a ser utilizado no sistema **deve conter o dicionário de seu próprio conteúdo** (V.T.O.C.), como foi mencionado anteriormente.

- Tipicamente, esse V.T.O.C. (dicionário) é uma tabela que, entre outras informações, registra as seguintes para cada arquivo :
  - **Número do Arquivo**
    (para uso interno pelo Sistema Operacional)
  - **Nome do Arquivo**
    (usado nas referências feitas pelos usuários ao arquivo)
  - **Comprimento do Registro Lógico** do arquivo
  - **Número total de registros lógicos** do arquivo
  - **Endereço físico** do 1º registro do arquivo, no volume
  - **Informações de proteção**
    (livre, acessível, *read only*, de acesso seletivo)

- Se o **acesso** ao arquivo for **seletivo**, há necessidade de se especificar o conjunto dos usuários que têm o direito de efetuar acessos, e que tipos de acessos pode cada um deles efetuar sobre o arquivo.
- Isso é implementado através de uma lista de informações relacionando os **usuários e os acessos permitidos** para cada um.

- Arquivos cujo **acesso** lógico é **sequencial** nem sempre são armazenados sequencialmente no volume de armazenamento, podendo estar **espalhados fisicamente** pelo mesmo, de acordo com algum critério determinado pelo sistema operacional, como por exemplo, a adoção de algum esquema de **alocação paginada** para os arquivos.
- Em outras palavras, nem sempre o arranjo lógico de um arquivo se reflete no arranjo físico do mesmo no volume em que está armazenado.

- Durante a execução do comando de entrada/saída acima citado, ocorrem normalmente os seguintes eventos:
  - **Procurar o nome** ARQ no dicionário (*diretório*) ou V.T.O.C. em que está contido, obtendo-se um número (que indica sua posição no dicionário);
  - **Obter** na linha do dicionário, correspondente a esse número, as **informações sobre o arquivo** ARQ;
  - Com base nas informações de proteção, aí obtidas, **decidir se o acesso solicitado deve ser permitido.**
  - A partir do endereço lógico R solicitado, **calcular o endereço lógico** (relativo ao início do arquivo), em *bytes*, do primeiro *byte* do registro R desejado, no arquivo.
  - **Transformar** o endereço lógico obtido **em um endereço físico** do primeiro *byte* do registro, no formato de par ordenado: <número do bloco, posição dentro do bloco>;
  - **Transferir o conteúdo do bloco** para uma área (*buffer*) auxiliar.
  - **Obter**, no bloco lido, **o registro lógico desejado**, e movê-lo para a posição de memória X (indicada pelo usuário) e seguintes.

Manutenção dos Dicionários dos Arquivos

- Por razões de economia de memória, os dicionários (*diretórios*) dos arquivos são usualmente **mantidos no disco**, sendo tratados, eles próprios, como se fossem arquivos normais, o que se justifica, uma vez que a maioria deles, sendo relativamente grandes, ocuparia excessivamente a memória principal se aí fosse mantido residente.

- **Acessos** a dicionários **em disco** são obviamente lentos, e tornam **ineficientes** os acessos aos arquivos, principalmente se forem frequentes.
- Por isso, considerando que apenas poucos arquivos são usados simultaneamente, o Sistema Operacional mantém **na memória**, para dicionários extensos, cópias das informações referentes aos arquivos em uso, em cada momento.

- Assim, tais elementos costumam ser copiados do disco para a memória, sendo aí mantidos **apenas enquanto** os arquivos associados **estiverem sendo usados.**
- A criação e manutenção dessa tabela parcial, que visa economizar tempo de acesso ao dicionário, utiliza duas **chamadas de supervisor – OPEN e CLOSE –** disparadas pelo programa que utiliza os arquivos envolvidos.
- Detalham-se adiante os módulos do sistema operacional que compõem o sistema de arquivos.

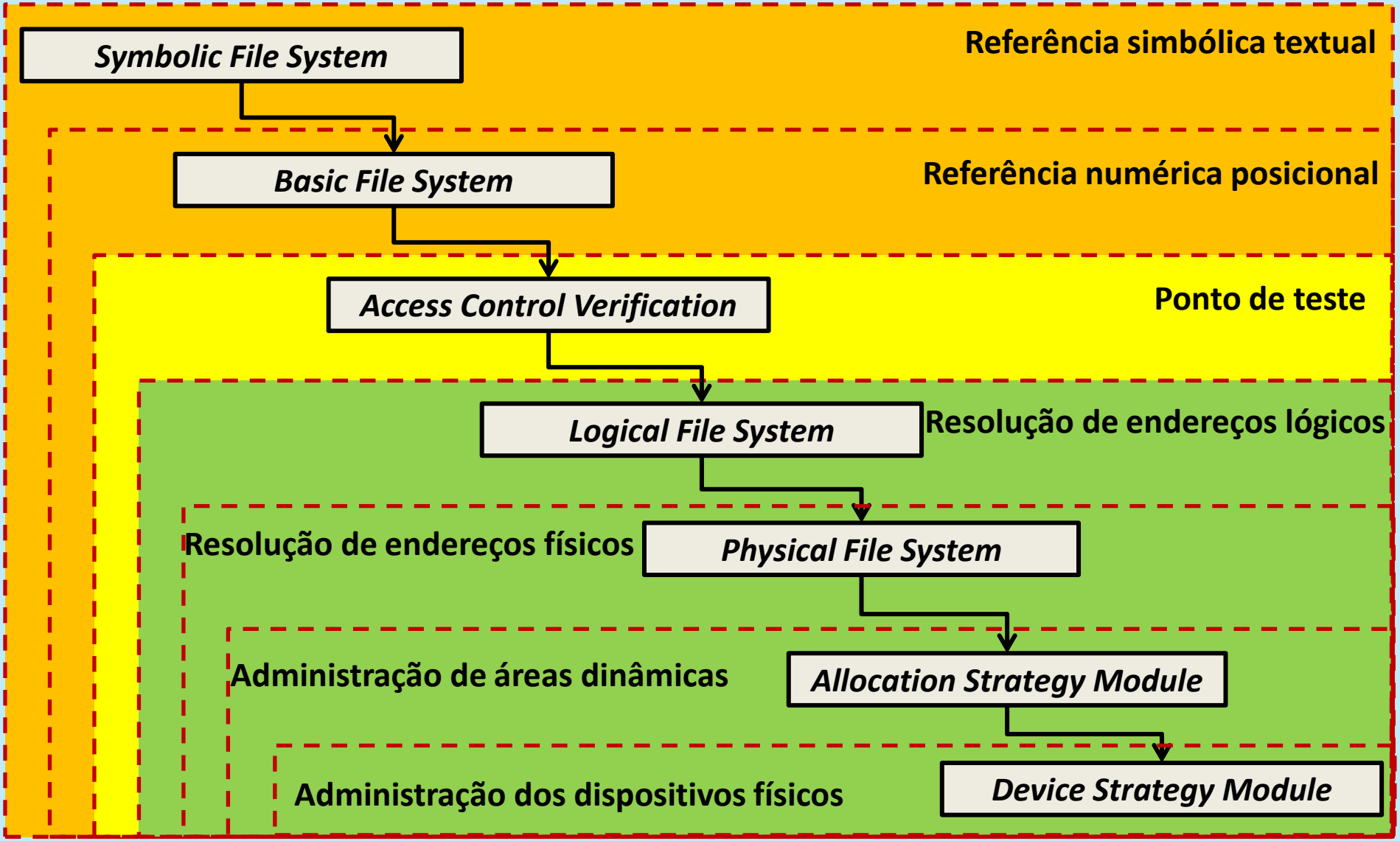
Função OPEN

- Antes de ser efetuada pela primeira vez a utilização de um arquivo, o programa deve solicitar ao sistema a execução da função **OPEN** para aquele arquivo.
- Esta efetua um acesso, no dicionário principal em disco, ao elemento a ele correspondente, e a transfere para a tabela dos arquivos ativos **AFT** (***active file table***), contida na memória.

Função CLOSE

- Depois que o arquivo for utilizado pela última vez pelo programa, torna-se desnecessária a sua presença na tabela de arquivos ativos, residente na memória.
- O programa deve então solicitar ao sistema a execução da função **CLOSE** para que tal informação seja regravada no disco, atualizando assim o dicionário principal.

Detalhamento dos componentes



Este módulo reúne todos os elementos referentes à nomenclatura de acesso aos arquivos por meio de referências simbólicas.

SFS - SYMBOLIC FILE SYSTEM

SFS (*Symbolic File System*)

- Este módulo é ativado indiretamente pelo programa do usuário através de uma chamada de supervisor referente ao código, gerado em linguagem de máquina, proveniente da tradução dos comandos de Entrada/Saída da linguagem de alto nível utilizada para codificar tal programa.

- Sua sequência de chamada tem usualmente uma estrutura da forma:

CALL SFS (<função>,<nome>,<registro>,<posição>)

- No caso particular do exemplo apresentado anteriormente, tem se:

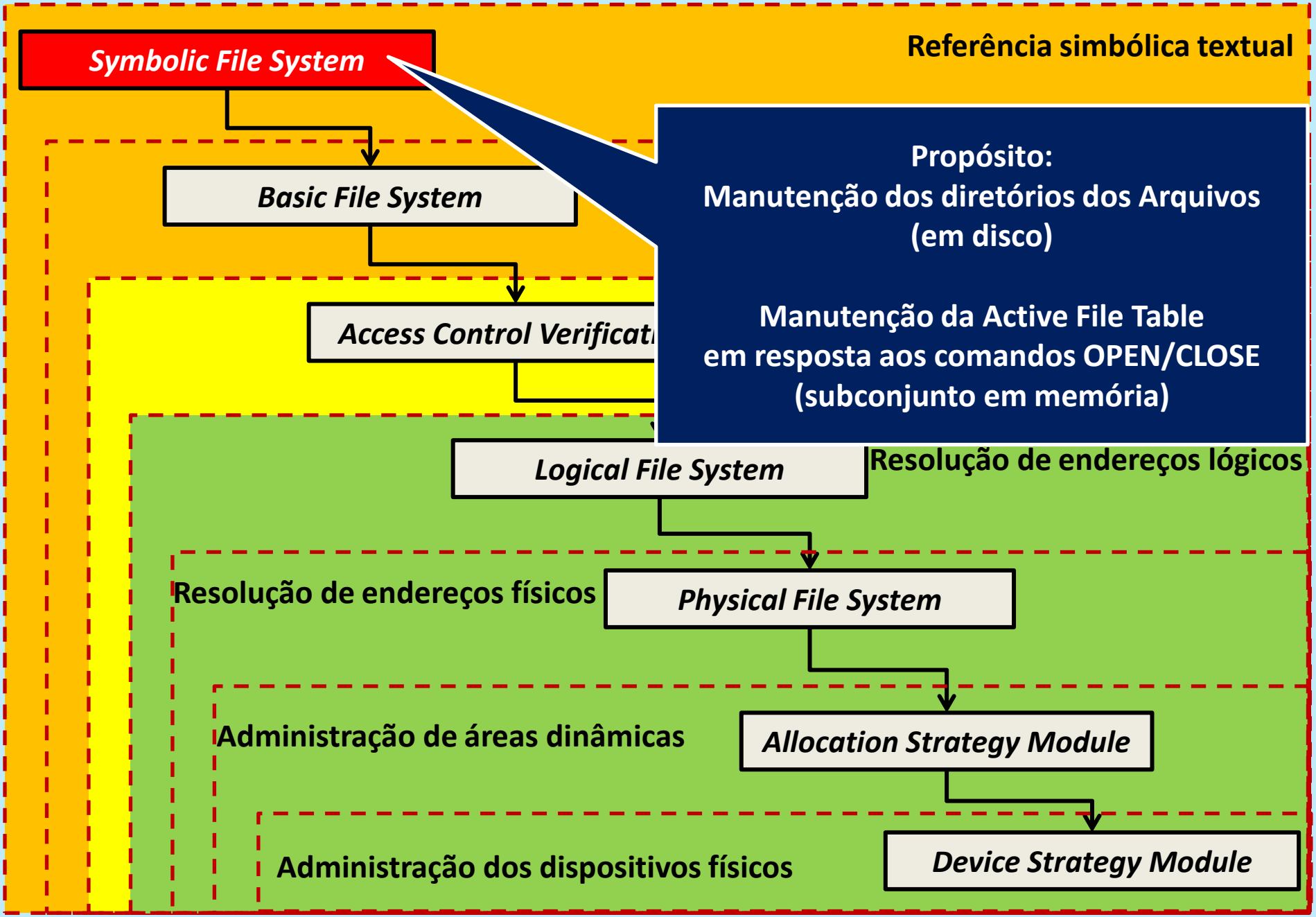
<função> = READ

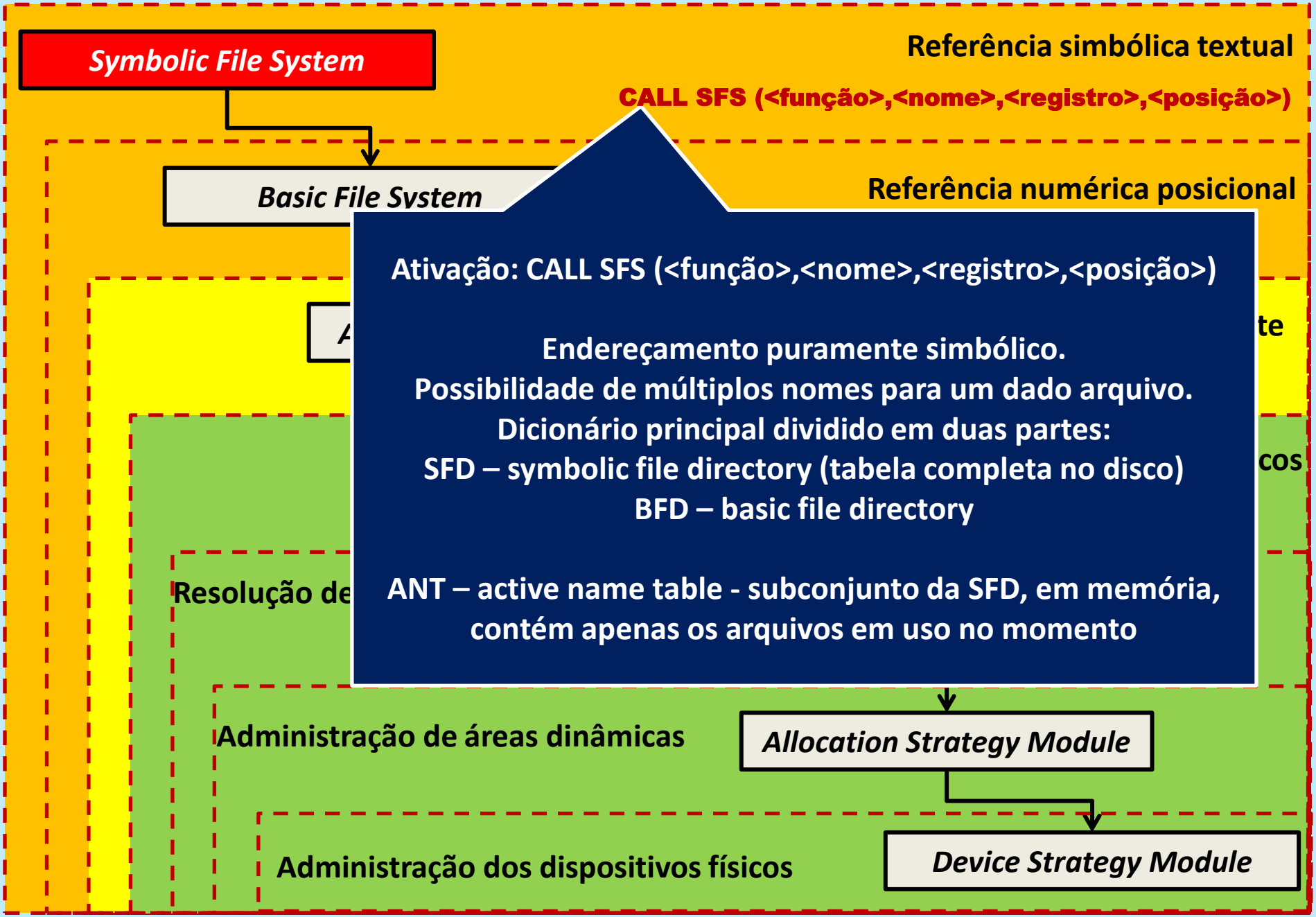
<nome> = "ARQ "

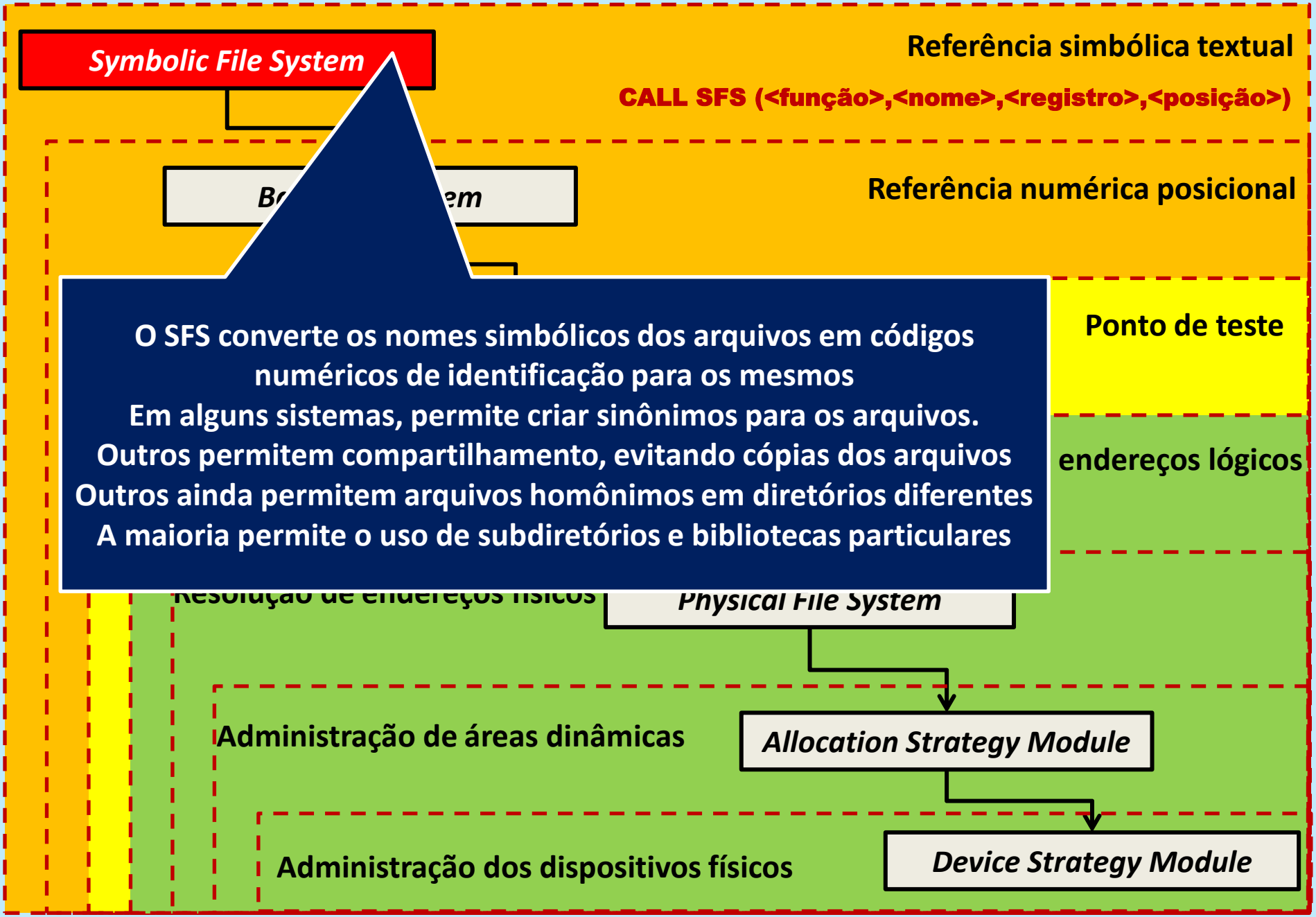
<registro> = R

<posição > = X

- Como se pode observar, neste nível de acesso ao arquivo, o endereçamento é **puramente simbólico**, pois provém da linguagem de alto nível, que é projetada para ser compreensível pelo programador.
- Em **sistemas simples**, existe uma correspondência **biunívoca** entre os <nomes> fornecidos pelo usuário e os arquivos existentes no sistema.
- Em **sistemas mais avançados**, pode-se ter o mesmo nome para **vários arquivos**, podendo-se também, se necessário, referenciar um mesmo arquivo físico por **diferentes nomes**.


Symbolic File System
Referência simbólica textual
Propósito:
Manutenção dos diretórios dos Arquivos
(em disco)
Manutenção da Active File Table
em resposta aos comandos OPEN/CLOSE
(subconjunto em memória)
Basic File System
Access Control Verificat
Logical File System
Resolução de endereços lógicos
Resolução de endereços físicos
Physical File System
Administração de áreas dinâmicas
Allocation Strategy Module
Administração dos dispositivos físicos
Device Strategy Module


Symbolic File System
Referência simbólica textual
CALL SFS (<função>,<nome>,<registro>,<posição>)
Basic File System
Referência numérica posicional
Ativação: CALL SFS (<função>,<nome>,<registro>,<posição>)
Endereçamento puramente simbólico.
Possibilidade de múltiplos nomes para um dado arquivo.
Dicionário principal dividido em duas partes:
SFD – symbolic file directory (tabela completa no disco)
BFD – basic file directory
ANT – active name table - subconjunto da SFD, em memória, contém apenas os arquivos em uso no momento
Resolução de
Administração de áreas dinâmicas
Allocation Strategy Module
Administração dos dispositivos físicos
Device Strategy Module



O SFS converte os nomes simbólicos dos arquivos em códigos numéricos de identificação para os mesmos
Em alguns sistemas, permite criar sinônimos para os arquivos.
Outros permitem compartilhamento, evitando cópias dos arquivos
Outros ainda permitem arquivos homônimos em diretórios diferentes
A maioria permite o uso de subdiretórios e bibliotecas particulares

- O dicionário principal é dividido em duas partes, por comodidade de manipulação:
  - **SFD** (*Symbolic File Directory*), que contém o nome dos arquivos, e a posição ID, na BFD, da informação referente ao arquivo.
    SFD constitui, portanto, uma **tabela de mapeamento** dos nomes dos arquivos para números que os identificam univocamente no sistema.
  - **BFD** (*Basic File Directory*), que contém todas as **informações complementares associadas aos** arquivos.

Arquivos de dicionário

- O SFD e o BFD podem ser representados separadamente em duas tabelas:

SFD

	Nome	ID
1	A	3
2	B	5
3	C	6

BFD

	Compr. do Reg. Lóg.	Núm. de reg. lóg.	End. Fís. 1o. bloco	Contr. acesso
1	-	-	0	BFD
2	20	3	1	SFD
3	80	10	2	Acesso permitido
4	1000	3	3	livre
5	500	7	6	Read only
6	100	30	10	A-read B- R/W
7	1000	2	12	livre
8	1000	1	15	livre

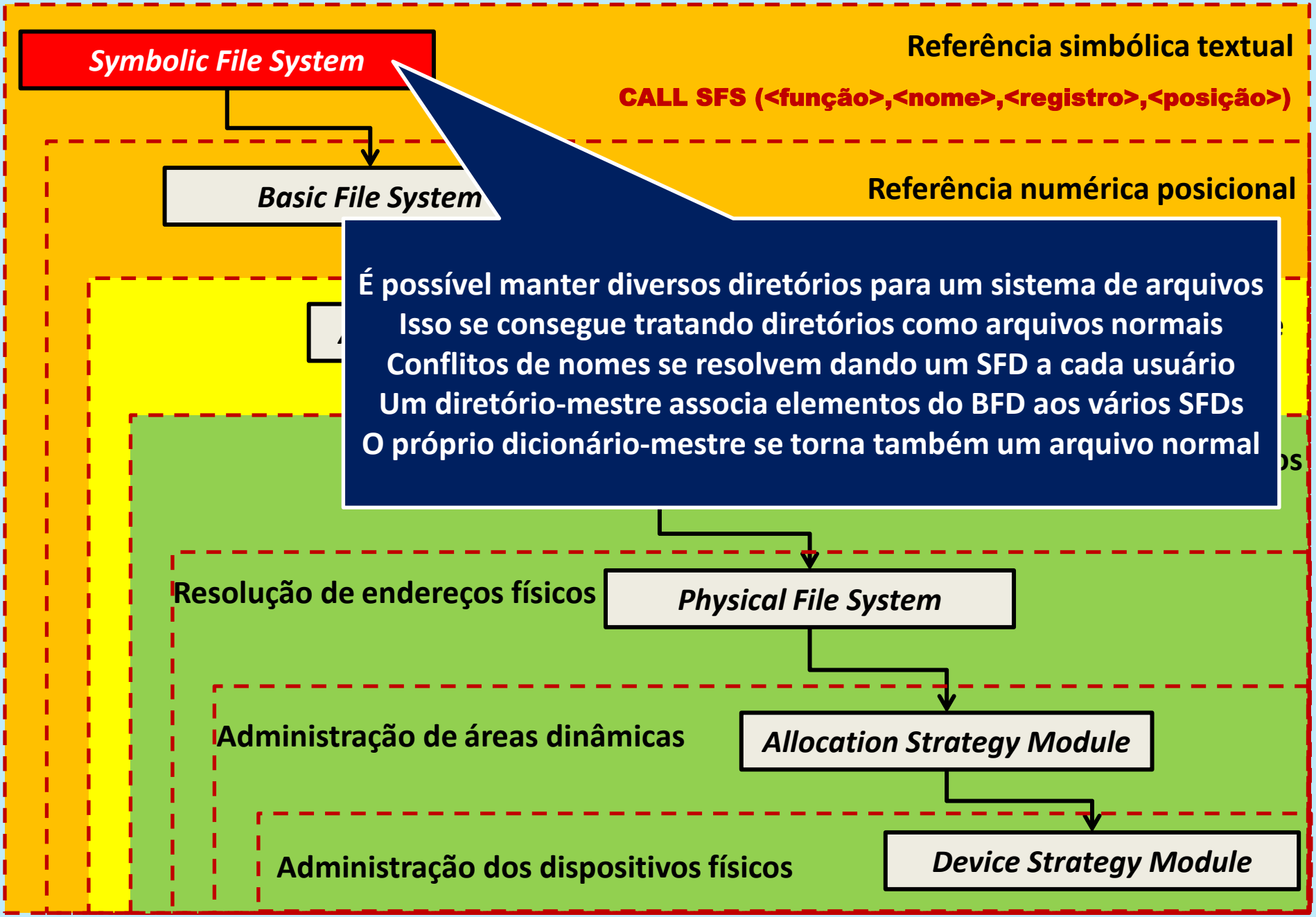
- SFD reside **no disco**, sendo trazidas à memória somente cópias dos elementos dessa tabela, associados aos arquivos correntemente em uso.

.

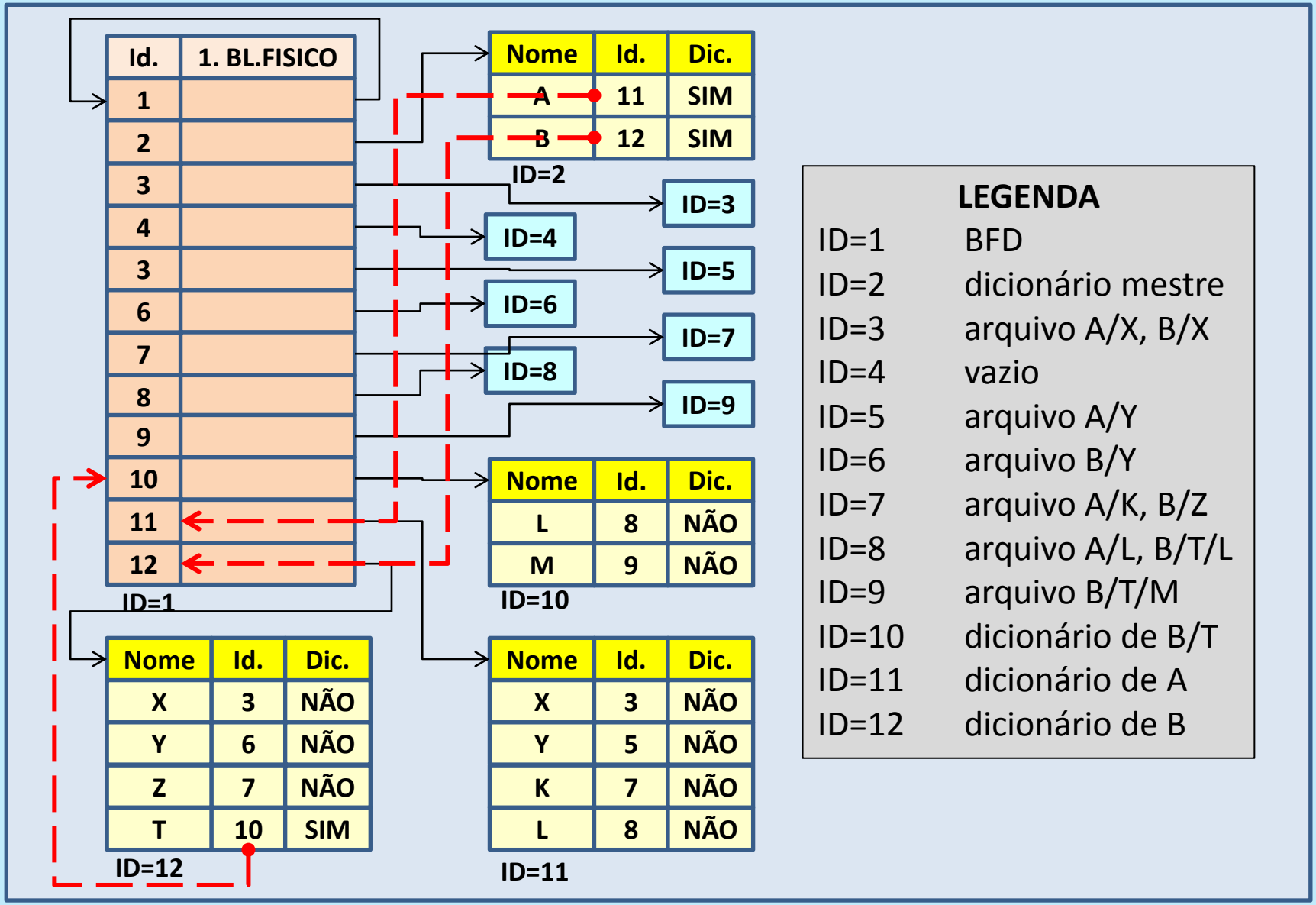
- A subtabela, assim formada, é residente na memória, e se chama ***Active Name Table*** **(ANT).**

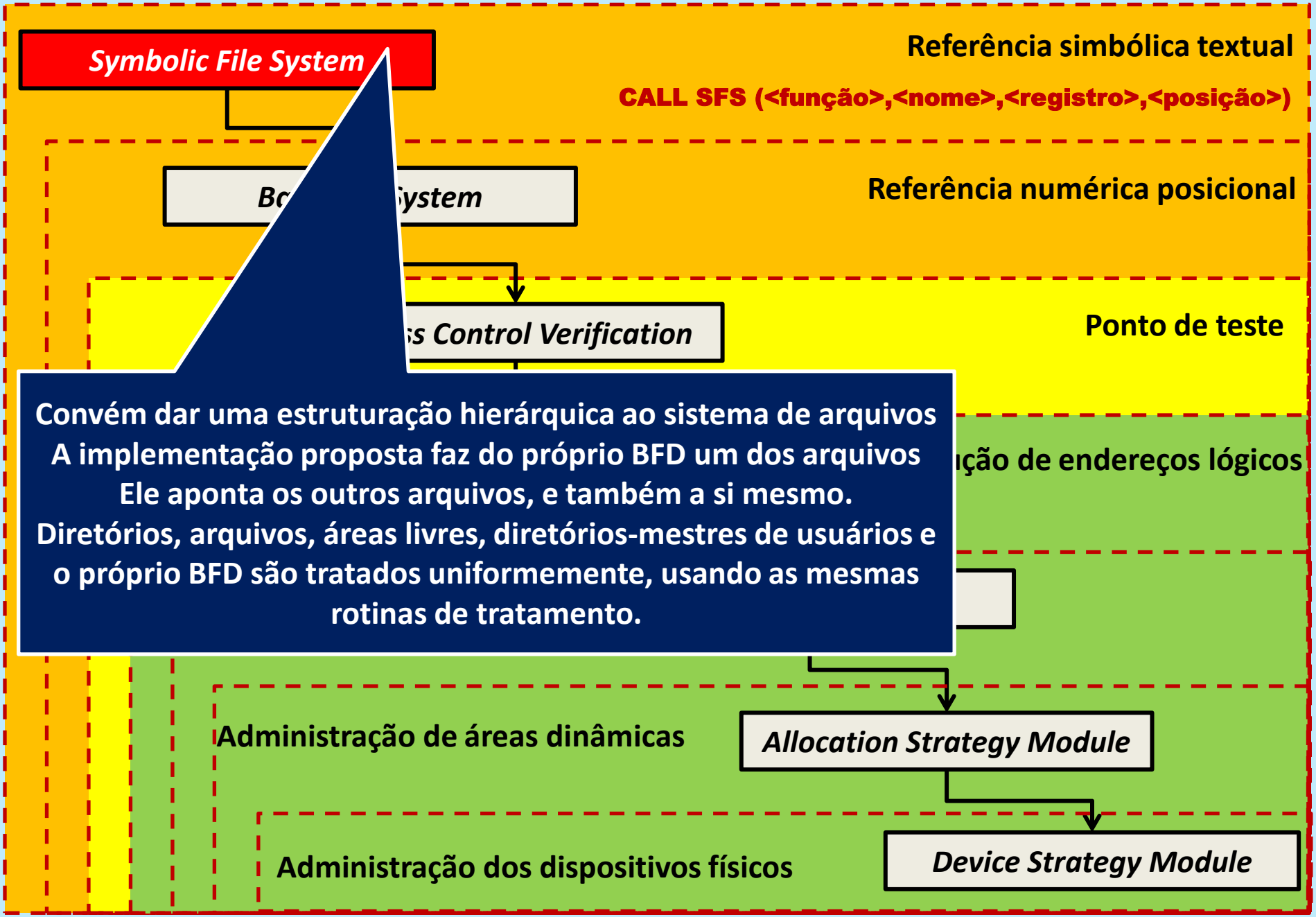
.

- SFS é responsável por manter as tabelas SFD e ANT, e se encarrega da comunicação com o módulo seguinte (BFS).


Symbolic File System
Referência simbólica textual
CALL SFS (<função>,<nome>,<registro>,<posição>)
Basic File System
Referência numérica posicional
É possível manter diversos diretórios para um sistema de arquivos
Isso se consegue tratando diretórios como arquivos normais
Conflitos de nomes se resolvem dando um SFD a cada usuário
Um diretório-mestre associa elementos do BFD aos vários SFDs
O próprio dicionário-mestre se torna também um arquivo normal
Resolução de endereços físicos
Physical File System
Administração de áreas dinâmicas
Allocation Strategy Module
Administração dos dispositivos físicos
Device Strategy Module

Resulta um sistema de arquivos hierárquico




Symbolic File System
Referência simbólica textual
CALL SFS (<função>,<nome>,<registro>,<posição>)
Referência numérica posicional
Ponto de teste
Convém dar uma estruturação hierárquica ao sistema de arquivos
A implementação proposta faz do próprio BFD um dos arquivos
Ele aponta os outros arquivos, e também a si mesmo.
Diretórios, arquivos, áreas livres, diretórios-mestres de usuários e o próprio BFD são tratados uniformemente, usando as mesmas rotinas de tratamento.
Administração de áreas dinâmicas
Allocation Strategy Module
Administração dos dispositivos físicos
Device Strategy Module

BFS – BASIC FILE SYSTEM

BFS (*Basic File System*)

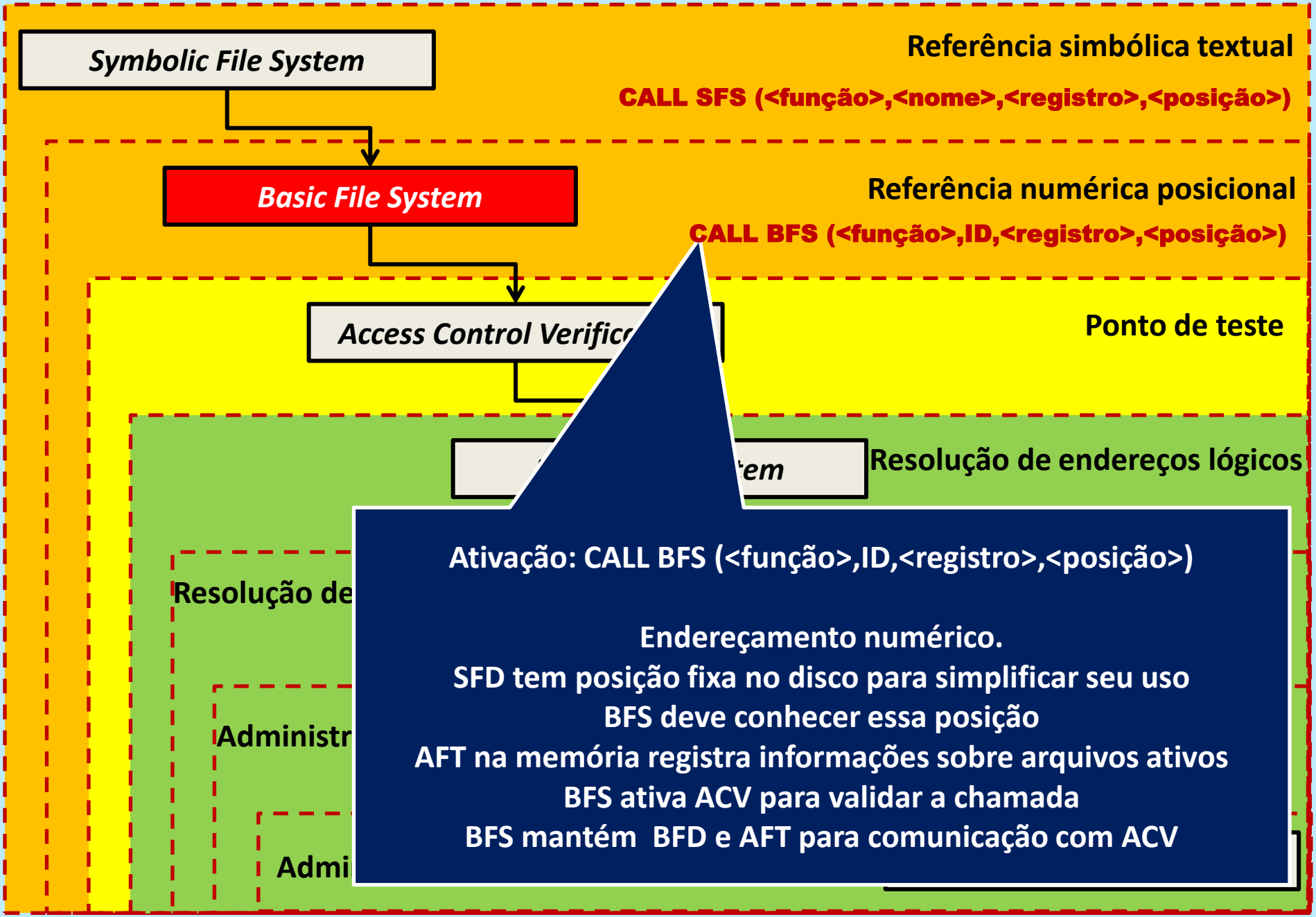
- Este módulo é ativado pelo SFS, através de uma chamada de procedimento, usualmente da seguinte forma:

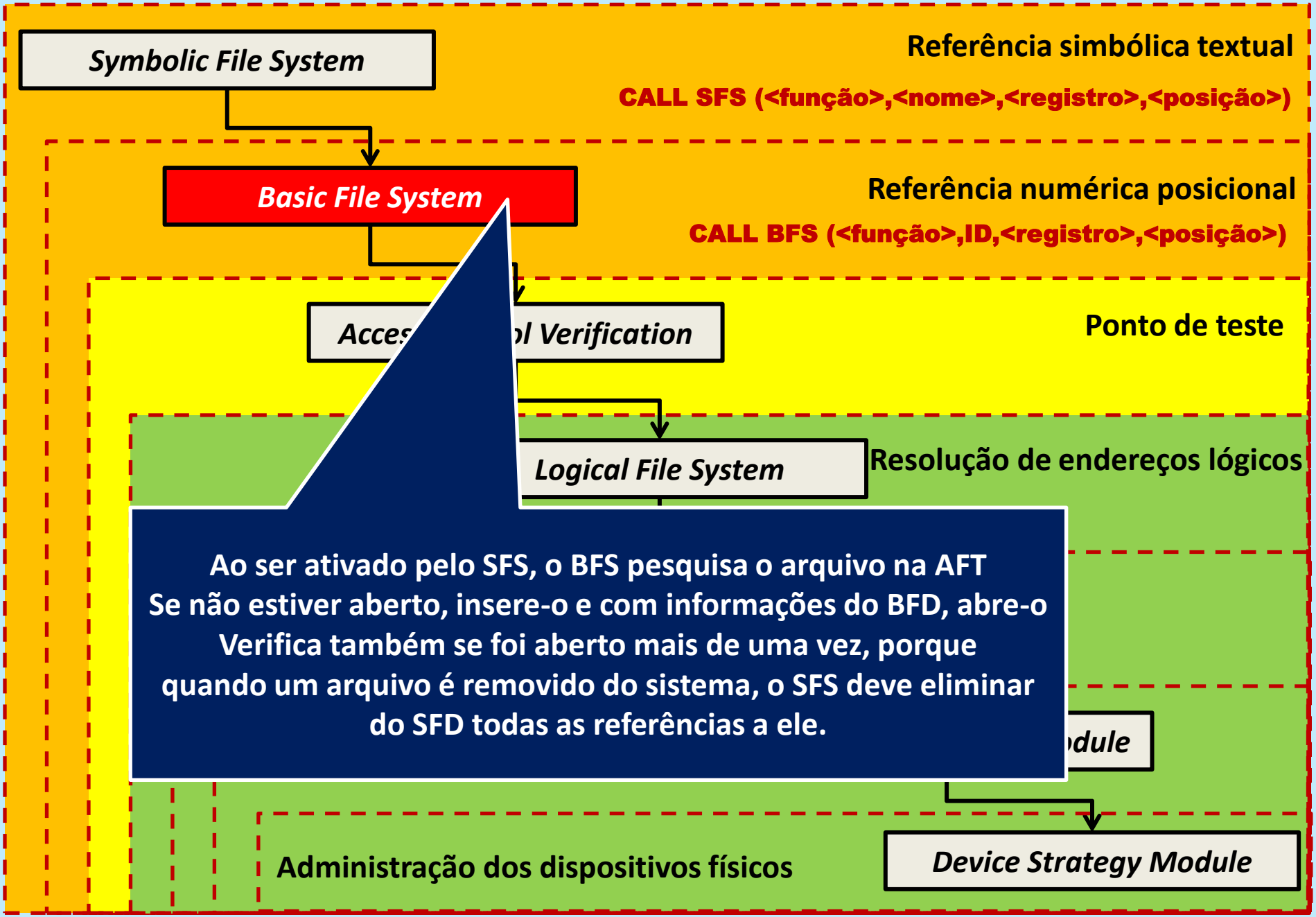
  CALL BFS (<função>,ID,<registro>,<posição>)

  onde ID é a posição das informações referentes ao arquivo desejado, no BFD.

- Deve localizar a posição ID do BFD, e acessar suas informações copiando-as na memória, para acesso mais rápido.

- O SFD é em geral alocado, portanto referenciado, em **posição fixa, conhecida, do disco**.
- O BFS deve ter conhecimento de tal posição para que possa efetuar o acesso ao mesmo.
- Na memória, a tabela que contém as informações sobre os arquivos em uso, retiradas do BFD, é denominada ***Active File Table*** **(AFT).**
- O BFS ativa o módulo ACV para a verificação da validade da chamada, substituindo o nome do arquivo ou seu ID pela posição da AFT, a ele associada.
- O BFS é responsável por manter as tabelas BFD e AFT, e pela comunicação com o módulo de verificação ACV.

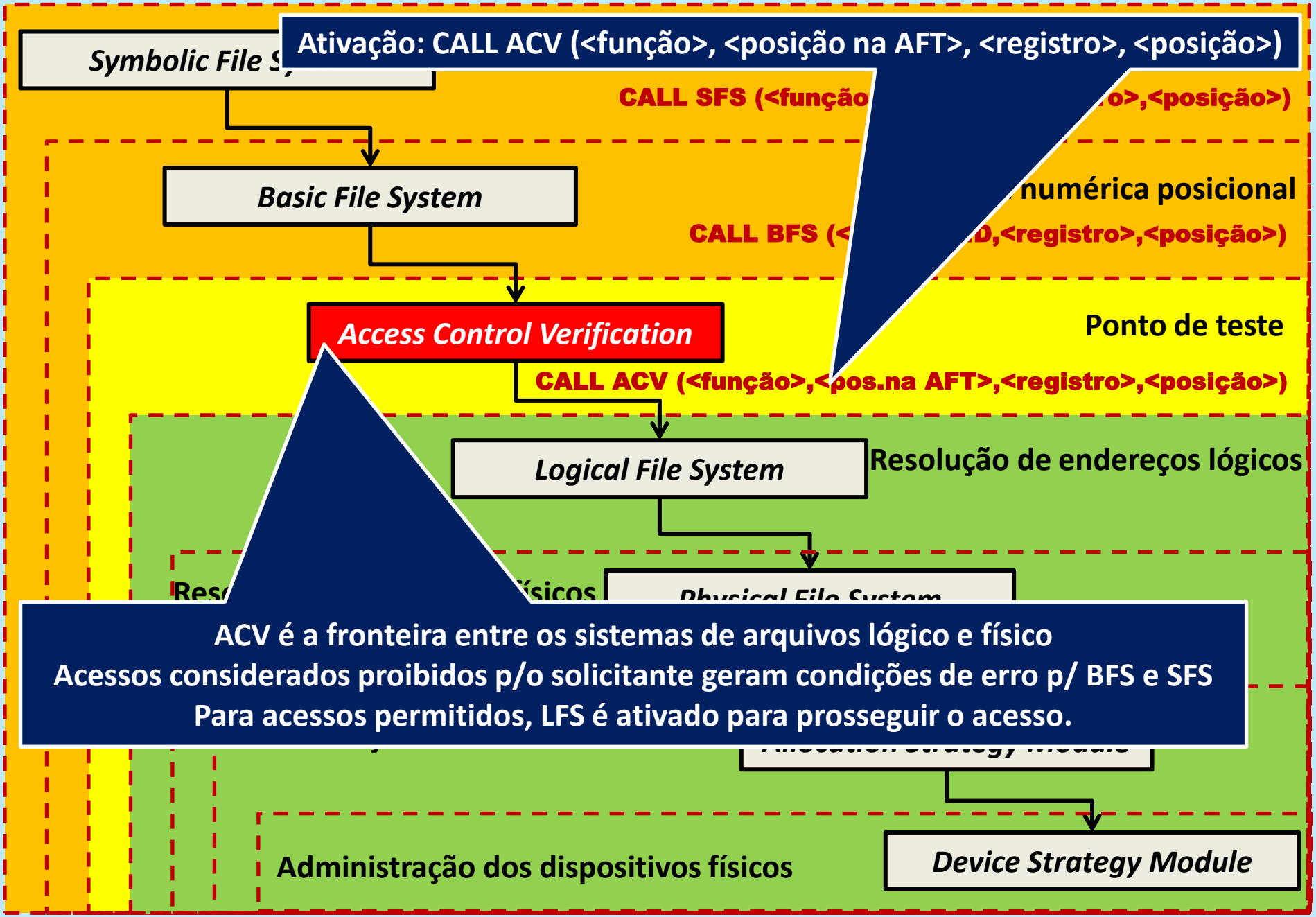

Symbolic File System
Referência simbólica textual
CALL SFS (<função>,<nome>,<registro>,<posição>)
Basic File System
Referência numérica posicional
CALL BFS (<função>,ID,<registro>,<posição>)
Access Control Verifico
Ponto de teste
em
Resolução de endereços lógicos
Resolução de
Administr
Admi
Ativação: CALL BFS (<função>,ID,<registro>,<posição>)
Endereçamento numérico.
SFD tem posição fixa no disco para simplificar seu uso
BFS deve conhecer essa posição
AFT na memória registra informações sobre arquivos ativos
BFS ativa ACV para validar a chamada
BFS mantém BFD e AFT para comunicação com ACV

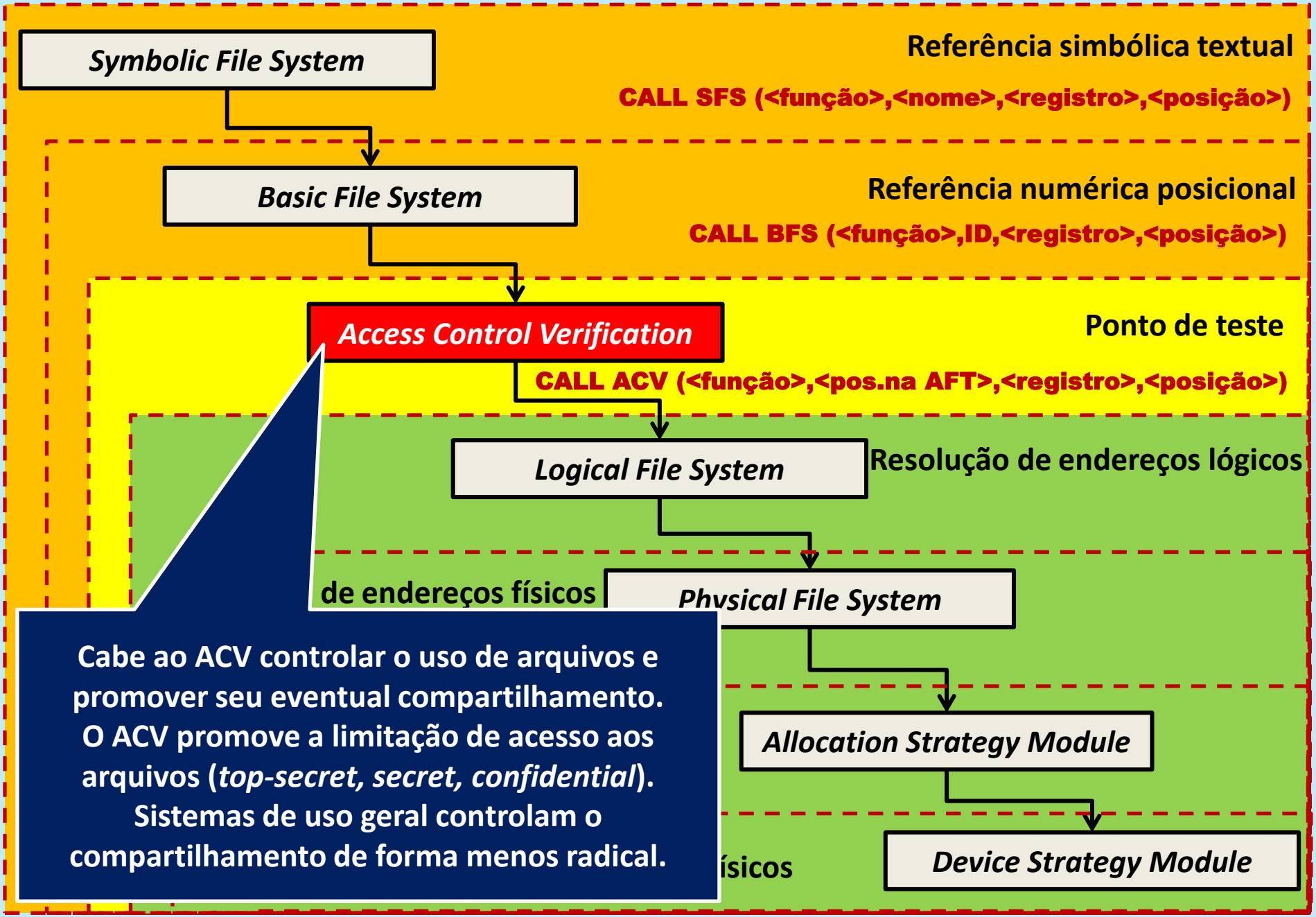

Symbolic File System
Referência simbólica textual
CALL SFS (<função>,<nome>,<registro>,<posição>)
Basic File System
Referência numérica posicional
CALL BFS (<função>,ID,<registro>,<posição>)
Ponto de teste
Logical File System
Resolução de endereços lógicos
Ao ser ativado pelo SFS, o BFS pesquisa o arquivo na AFT
Se não estiver aberto, insere-o e com informações do BFD, abre-o
Verifica também se foi aberto mais de uma vez, porque
quando um arquivo é removido do sistema, o SFS deve eliminar
do SFD todas as referências a ele.
Administração dos dispositivos físicos
Device Strategy Module

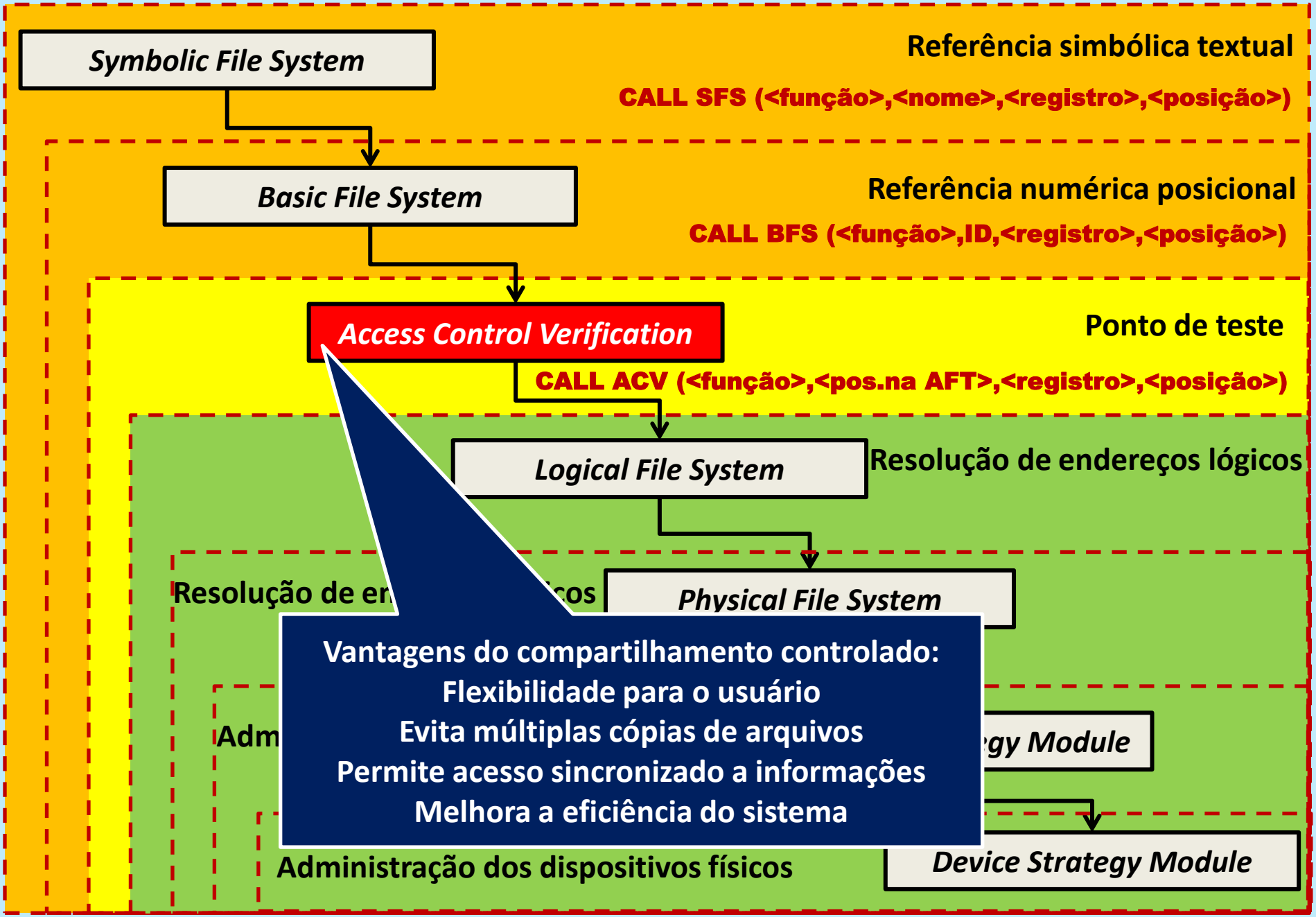
ACV – ACCESS CONTROL VERIFICATION

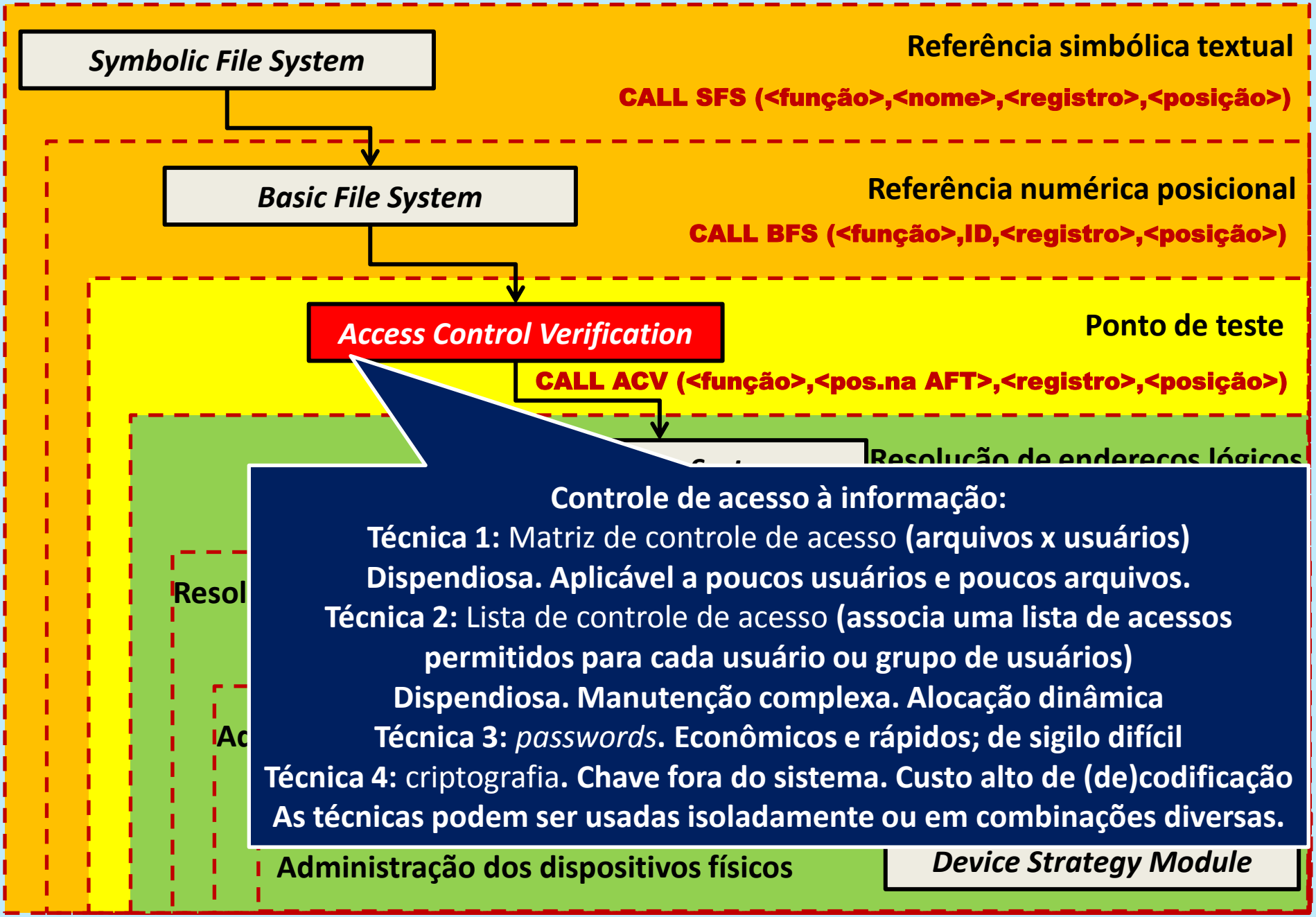
ACV (*Access Control Verification Module*)

- É ativado pelo BFS, da seguinte forma: CALL ACV (<função>, <posição na AFT>, <registro>, <posição>)
- É o **ponto de teste**, é a **fronteira** entre os sistemas físico e lógico (BFS, LFS)
- Se o acesso ao arquivo não for considerado permitido para o usuário solicitante, uma condição de erro é devolvida aos módulos anteriores (BFS, SFS), e o programa do usuário é descontinuado pelo sistema operacional, ou então simplesmente informado acerca do erro detectado.
- Se o acesso ao arquivo for considerado permitido, o LFS será ativado, para prosseguir o acesso ao arquivo.


Ativação: CALL ACV (<função>, <posição na AFT>, <registro>, <posição>)
Symbolic File S
CALL SFS (<função
o>,<posição>)
Basic File System
numérica posicional
CALL BFS (<
D,<registro>,<posição>)
Access Control Verification
Ponto de teste
CALL ACV (<função>,<pos.na AFT>,<registro>,<posição>)
Logical File System
Resolução de endereços lógicos
Res
ísicos
Physical File System
ACV é a fronteira entre os sistemas de arquivos lógico e físico
Acessos considerados proibidos p/o solicitante geram condições de erro p/ BFS e SFS
Para acessos permitidos, LFS é ativado para prosseguir o acesso.
Allocation Strategy Module
Administração dos dispositivos físicos
Device Strategy Module


Symbolic File System
Referência simbólica textual
CALL SFS (<função>,<nome>,<registro>,<posição>)
Basic File System
Referência numérica posicional
CALL BFS (<função>,ID,<registro>,<posição>)
Access Control Verification
Ponto de teste
CALL ACV (<função>,<pos.na AFT>,<registro>,<posição>)
Logical File System
Resolução de endereços lógicos
de endereços físicos
Physical File System
Allocation Strategy Module
ísicos
Device Strategy Module
Cabe ao ACV controlar o uso de arquivos e promover seu eventual compartilhamento. O ACV promove a limitação de acesso aos arquivos (*top-secret, secret, confidential*). Sistemas de uso geral controlam o compartilhamento de forma menos radical.


Symbolic File System
Referência simbólica textual
CALL SFS (<função>,<nome>,<registro>,<posição>)
Basic File System
Referência numérica posicional
CALL BFS (<função>,ID,<registro>,<posição>)
Access Control Verification
Ponto de teste
CALL ACV (<função>,<pos.na AFT>,<registro>,<posição>)
Logical File System
Resolução de endereços lógicos
Physical File System
Vantagens do compartilhamento controlado:
Flexibilidade para o usuário
Evita múltiplas cópias de arquivos
Permite acesso sincronizado a informações
Melhora a eficiência do sistema
Administração dos dispositivos físicos
Device Strategy Module


Symbolic File System
Referência simbólica textual
CALL SFS (<função>,<nome>,<registro>,<posição>)
Basic File System
Referência numérica posicional
CALL BFS (<função>,ID,<registro>,<posição>)
Access Control Verification
Ponto de teste
CALL ACV (<função>,<pos.na AFT>,<registro>,<posição>)
Resolução de endereços lógicos
Resol
Ac
Controle de acesso à informação:
Técnica 1: Matriz de controle de acesso (arquivos x usuários)
Dispendiosa. Aplicável a poucos usuários e poucos arquivos.
Técnica 2: Lista de controle de acesso (associa uma lista de acessos permitidos para cada usuário ou grupo de usuários)
Dispendiosa. Manutenção complexa. Alocação dinâmica
Técnica 3: passwords. Econômicos e rápidos; de sigilo difícil
Técnica 4: criptografia. Chave fora do sistema. Custo alto de (de)codificação
As técnicas podem ser usadas isoladamente ou em combinações diversas.
Administração dos dispositivos físicos
Device Strategy Module

LFS – LOGICAL FILE SYSTEM

LFS (*Logical File System*)

- É ativado pelo ACV, no caso de a chamada ser válida:

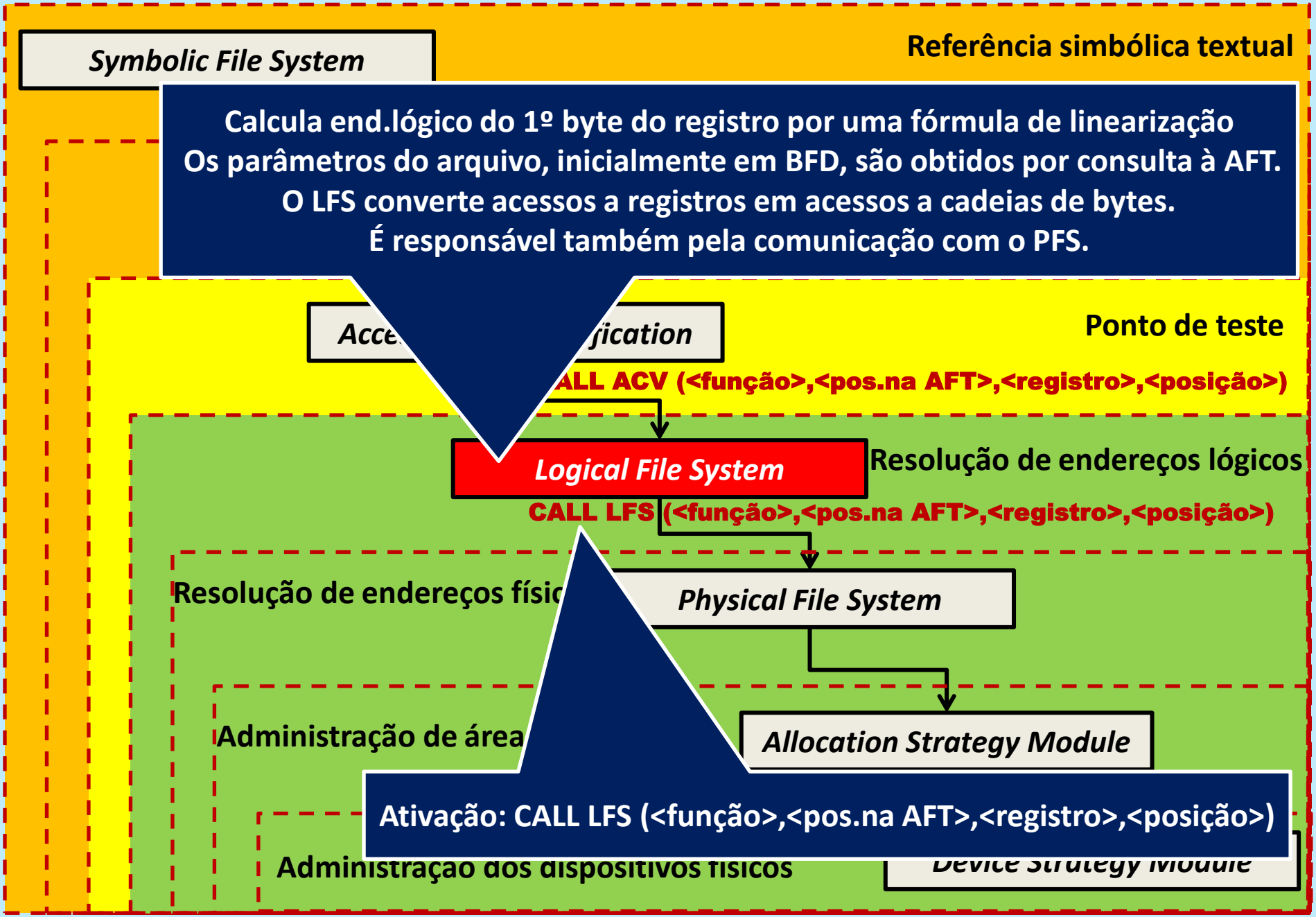
CALL LFS (<função>, <posição na AFT>, <registro>, <posição>)

- Converte o pedido de transferência de um registro para pedido de transferência de uma cadeia de *bytes*.
- Calcula o <endereço lógico> do primeiro *byte* do <registro> através de uma fórmula de linearização:

< endereço físico do primeiro *byte* do arquivo> + <número do registro>*<comprimento do registro>

onde o <endereço lógico do primeiro *byte* do registro> é a posição relativa do primeiro *byte* do registro, com referência ao primeiro *byte* do arquivo.

- O <comprimento do registro> é o tamanho, em *bytes,* de um registro lógico, e é obtido através da consulta à copia da entrada correspondente ao arquivo na AFT (esta informação está contida originariamente na BFD).
- Obtém também na AFT o número de *bytes* a transferir (<comprimento lógico em *bytes*>), e que corresponde ao <comprimento do registro>.
- O LFS é responsável por:
  - converter pedidos de acesso a registros em pedidos de acesso a cadeias de *bytes.*
  - comunicação com o PFS.


Symbolic File System
Referência simbólica textual
Calcula end.lógico do 1º byte do registro por uma fórmula de linearização
Os parâmetros do arquivo, inicialmente em BFD, são obtidos por consulta à AFT.
O LFS converte acessos a registros em acessos a cadeias de bytes.
É responsável também pela comunicação com o PFS.
Ponto de teste
ALL ACV (<função>,<pos.na AFT>,<registro>,<posição>)
Logical File System
Resolução de endereços lógicos
CALL LFS (<função>,<pos.na AFT>,<registro>,<posição>)
Physical File System
Allocation Strategy Module
Ativação: CALL LFS (<função>,<pos.na AFT>,<registro>,<posição>)

O LFS mapeia a estrutura de registros lógicos dos arquivos em endereços relativos lineares das cadeias de bytes que compõem esses registros. Fornece várias opções de estruturação lógica de arquivos:
Sequencial, com registros de tamanho fixo, e acesso direto ou sequencial;
Sequencial, com registros de tamanho variável, e acesso preferencialmente sequencial.

(continua...)

Referência simbólica textual

CALL SFS (<função>,<nome>,<registro>,<posição>)

System

Referência numérica posicional

CALL BFS (<função>,ID,<registro>,<posição>)

ess Control Verification

Ponto de teste

CALL ACV (<função>,<pos.na AFT>,<registro>,<posição>)

Logical File System

Resolução de endereços lógicos

CALL LFS (<função>,<pos.na AFT>,<registro>,<posição>)

ereços físicos

Physical File System

ão de áreas dinâmicas

Allocation Strategy Module

Administração dos dispositivos físicos

Device Strategy Module

O LFS mapeia a estrutura de registros lógicos d arquivos em endere relativos linea cadeias de by compõem esses r Fornece várias op estruturação lógica de arquivos:
Sequencial, com registros de tamanho fixo, e acesso direto ou sequencial;
Sequencial, com registros de tamanho variável, e acesso preferencialmente sequencial.

(continua...)

O acesso direto neste caso é ineficiente.
Pode ser melhorado:
Usando o método de controle do elevador;
Usando tabelas que informam o comprimento de cada registro do arquivo;
Memorizando o comprimento dos registros já acessados e seus endereços

bólica textual

sição>)

ional

ção>)

Ponto de teste

CALL A pos.na AFT>,<registro>,<posição>)

Logical File System

Resolução de endereços lógicos

CALL LFS (<função>,<pos.na AFT>,<registro>,<posição>)

reços físicos

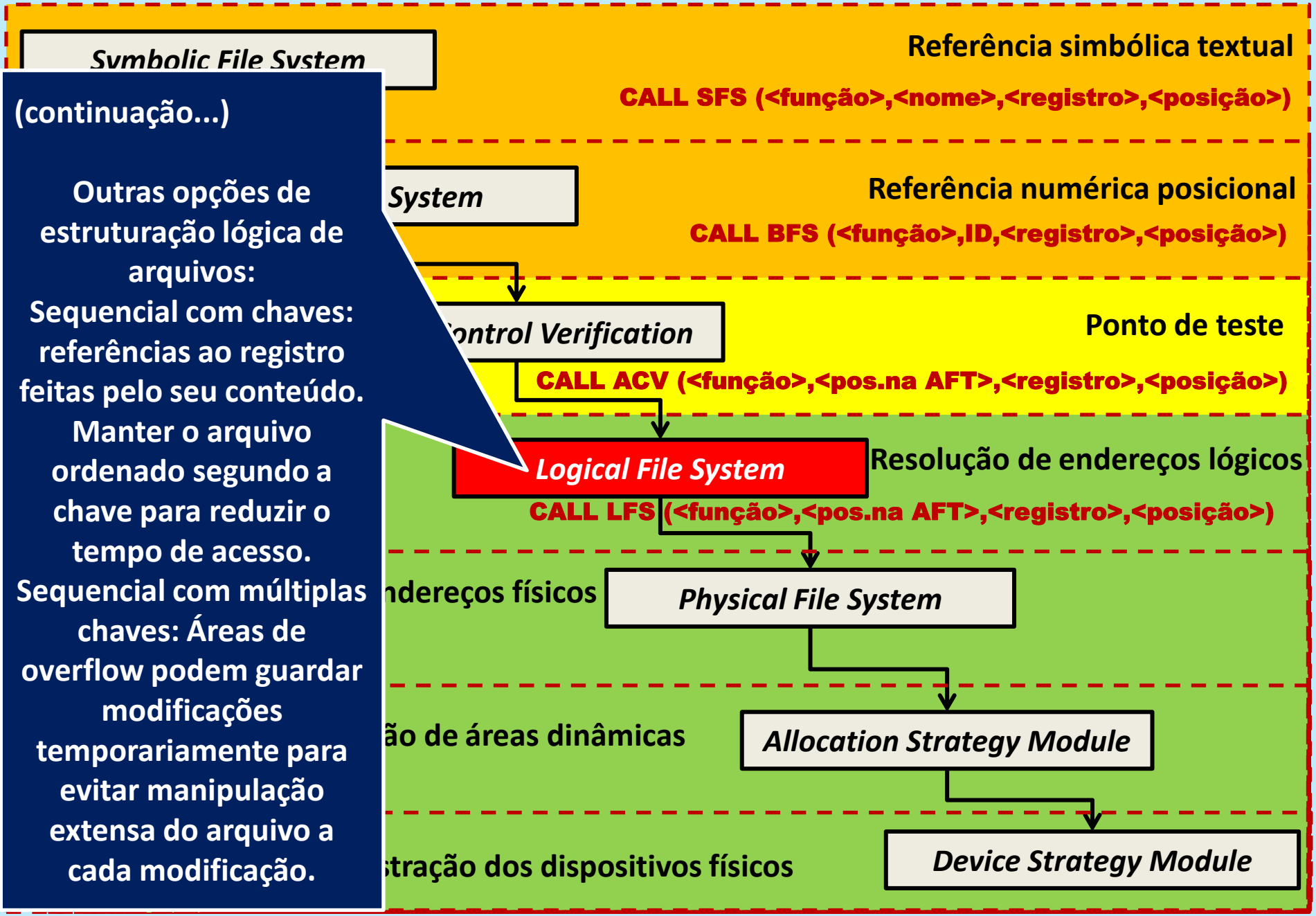
Physical File System

ão de áreas dinâmicas

Allocation Strategy Module

Administração dos dispositivos físicos

Device Strategy Module



(continuação...)

Outras opções de estruturação lógica de arquivos:
Sequencial com chaves: referências ao registro feitas pelo seu conteúdo. Manter o arquivo ordenado segundo a chave para reduzir o tempo de acesso.
Sequencial com múltiplas chaves: Áreas de overflow podem guardar modificações temporariamente para evitar manipulação extensa do arquivo a cada modificação.

PFS – PHYSICAL FILE SYSTEM

PFS (*Physical File System*)

- É ativado pelo LFS, da seguinte maneira:

  CALL PFS (<função>, <posição na AFT>, <endereço lógico do primeiro *byte* do registro>, <comprimento do registro em *bytes*>, <posição>).

- Deve calcular (mapear), a partir do <endereço lógico do primeiro *byte* do registro>, o <endereço físico do registro>, no formato de par ordenado (<bloco>, <posição do bloco>) onde <bloco> é, usualmente, o endereço do setor do disco, no qual se encontra a informação desejada, e <posição dentro do bloco> é a posição relativa do primeiro *byte* da informação desejada, com referência ao início do setor.

Mapeamento de endereços físicos

- Tratam-se, como se pode notar, de informações sobre a posição física da informação, e são calculadas através de fórmulas tais como:

$$< bloco >= int\left(\frac{< endereço\ lógico\ do\ primeiro\ byte\ do\ registro >}{< tamanho\ físico\ do\ bloco >}\right) + < endereço\ físico\ do\ primeiro\ bloco\ do\ arquivo >$$

e

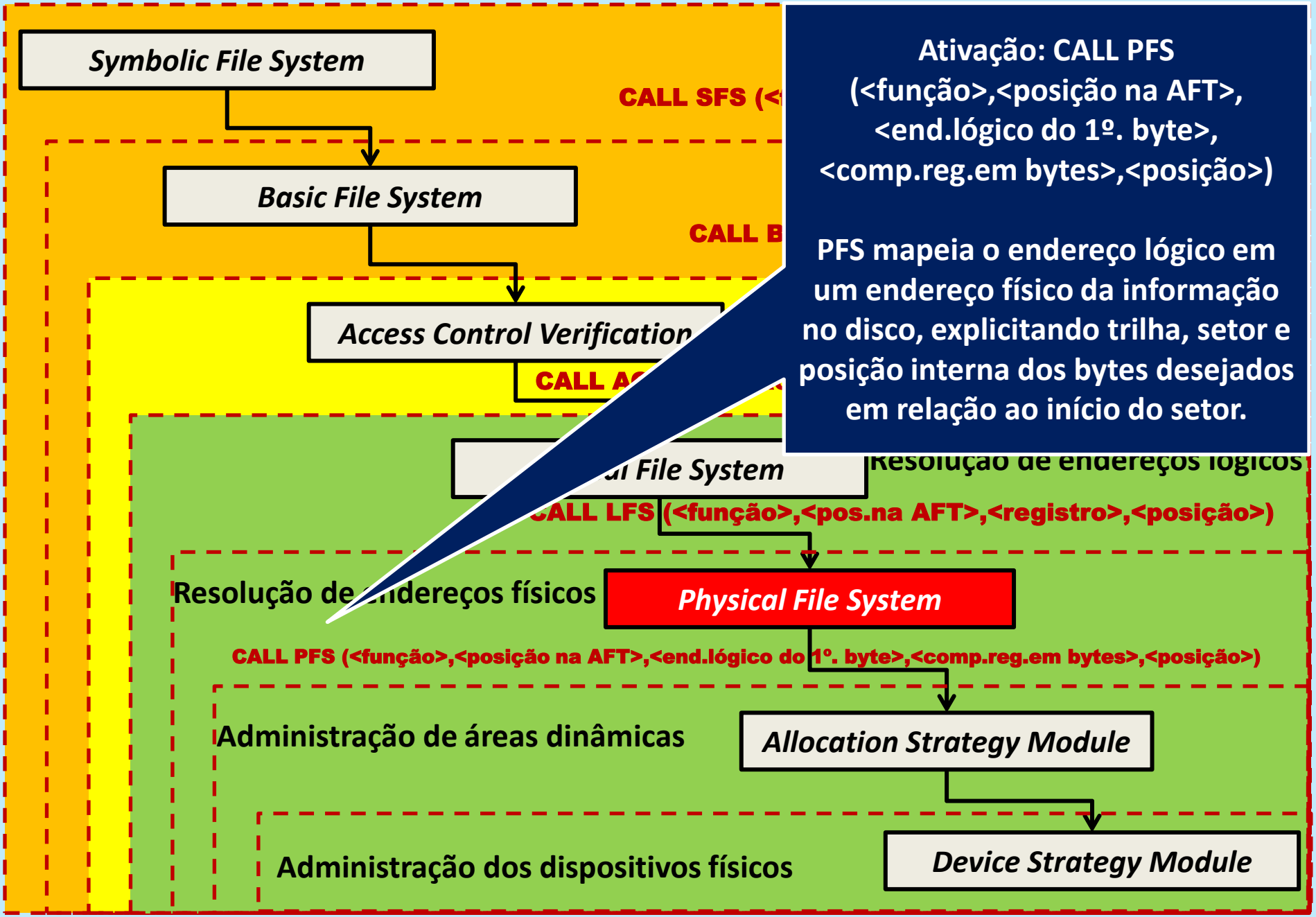
$$< posição\ dentro\ do\ bloco > = rem\ \left(\frac{< endereço\ lógico\ do\ primeiro\ byte\ do\ registro >}{< tamanho\ físico\ do\ bloco >}\right)$$

- Para que o PFS tenha condições de efetuar tais conversões, deverá, naturalmente, ter acesso às **leis de mapeamento** adotadas pelo sistema para cada dispositivo sob sua responsabilidade.
- Se os dispositivos forem diferentes entre si, é necessário que o PFS disponha de descritores para cada um dos **métodos de mapeamento** adotados.
- Tais informações estão organizadas na forma de uma tabela que é gerada por ocasião da iniciação do sistema, e que se denomina ***Physical Organization Table*** **(POT).**

- Sempre que a <função> for WRITE, e que a alocação de blocos físicos for dinâmica, se no instante da execução do PFS não estiver disponível para a gravação nenhum bloco físico já alocado para isso, o módulo de alocação dinâmica **ASM (*Allocation Strategy Module*)** deverá ser ativado pelo PFS, para ocupar-se de indicar o endereço de algum bloco livre, que servirá como destino da transferência de dados a ser realizada.

- Note-se a **analogia** que esse cenário guarda com o da **alocação dinâmica de memória**.
- Em sistemas mais simples, não existe alocação dinâmica, e portanto todos os blocos são alocados de uma só vez, e devolvidos ao sistema também de uma só vez, através de chamadas de supervisor "CREATE" e "DESTROY", que devem ser efetuadas pelo programa que acessa os arquivos.

- As funções do PFS são, portanto:
  - Converter pedidos de cadeia lógica de *bytes* em um par ordenado (<número do bloco>, <posição>).
  - Utilizar as tabelas POT para efetuar os **mapeamentos** de endereços
  - Comunicar-se com os módulos ASM e DSM (descritos adiante) quando necessário, para efetuar **alocação dinâmica** e para prosseguir a preparação da entrada/saída desejada, respectivamente.


Symbolic File System
CALL SFS (<
Basic File System
CALL B
Access Control Verification
CALL A
al File System
Resolução de endereços lógicos
CALL LFS (<função>,<pos.na AFT>,<registro>,<posição>)
Resolução de endereços físicos
Physical File System
CALL PFS (<função>,<posição na AFT>,<end.lógico do 1º. byte>,<comp.reg.em bytes>,<posição>)
Administração de áreas dinâmicas
Allocation Strategy Module
Administração dos dispositivos físicos
Device Strategy Module
Ativação: CALL PFS (<função>,<posição na AFT>, <end.lógico do 1º. byte>, <comp.reg.em bytes>,<posição>)
PFS mapeia o endereço lógico em um endereço físico da informação no disco, explicitando trilha, setor e posição interna dos bytes desejados em relação ao início do setor.

Symbolic File System

CALL SFS (<

Basic File System

CALL B

Estas operações costumam ser executadas com a ajuda de fórmulas específicas para cada dispositivo envolvido, e têm o aspecto mostrado abaixo:

Ativação: CALL PFS (<função>,<posição na AFT>, <end.lógico do 1º. byte>, <comp.reg.em bytes>,<posição>)

PFS mapeia o endereço lógico em um endereço físico da informação no disco, explicitando trilha, setor e posição interna dos bytes desejados em relação ao início do setor.

ção de endereços lógicos

L LFS (<função>,<posição na AFT>,<registro>,<posição>)

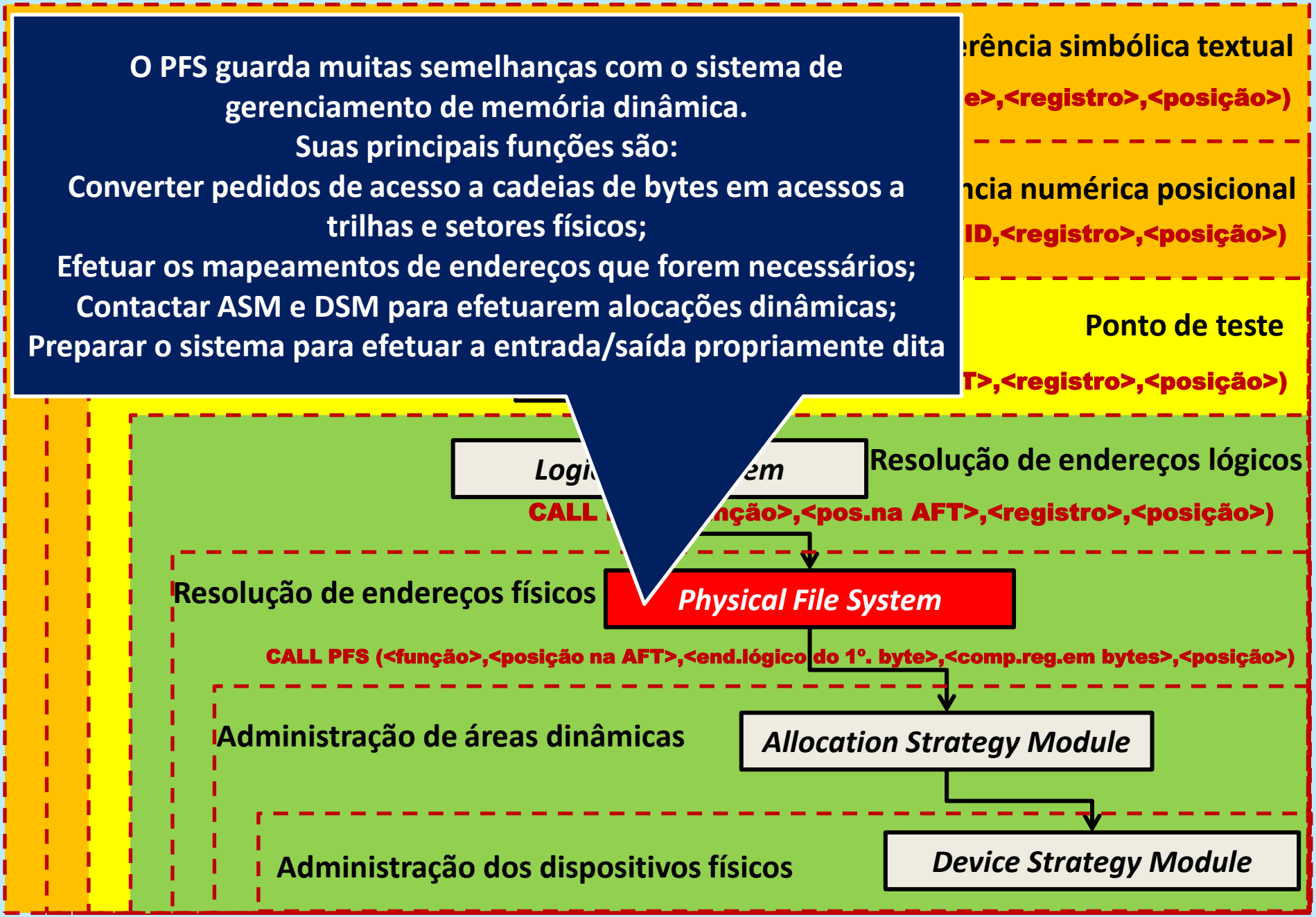
Resolução de endereços físicos

Physical File System

$$< bloco >= int\left(\frac{< endereço\ lógico\ do\ primeiro\ byte\ do\ registro >}{< tamanho\ físico\ do\ bloco >}\right) + < endereço\ físico\ do\ primeiro\ bloco\ do\ arquivo >$$

$$< posição\ dentro\ do\ bloco > = rem\ (\frac{< endereço\ lógico\ do\ primeiro\ byte\ do\ registro >}{< tamanho\ físico\ do\ bloco >})$$

O PFS guarda muitas semelhanças com o sistema de gerenciamento de memória dinâmica.
Suas principais funções são:
Converter pedidos de acesso a cadeias de bytes em acessos a trilhas e setores físicos;
Efetuar os mapeamentos de endereços que forem necessários;
Contactar ASM e DSM para efetuarem alocações dinâmicas;
Preparar o sistema para efetuar a entrada/saída propriamente dita



Uma função do PFS é a de minimizar operações físicas de entrada/saída
Uma técnica para isso é o uso de buffering: um ou mais buffers exclusivos ou buffer pool dinamicamente compartilhado entre os arquivos ativos.

Registros físicos e lógicos nem sempre têm tamanhos iguais. Isso se reflete na eficiência dos acessos ao arquivo.
O uso de tamanho variável de registro costuma ser dispendioso.
Técnicas usuais para tratar esse problema são os blocos ligados e os file maps, que propiciam a alocação não contígua de áreas para os arquivos.

Symboli

ual

>)

nal

>)

te

>)

ile System

Resolução de endereços lógicos

(<função>,<pos.na AFT>,<registro>,<posição>)

Resolução de endereços físicos

Physical File System

CALL PFS (<função>,<posição na AFT>,<end.lógico do 1º. byte>,<comp.reg.em bytes>,<posição>)

Administração de áreas dinâmicas

Allocation Strategy Module

Administração dos dispositivos físicos

Device Strategy Module

ASM – ALLOCATION STRATEGY MODULE

ASM (*Allocation Strategy Module*)

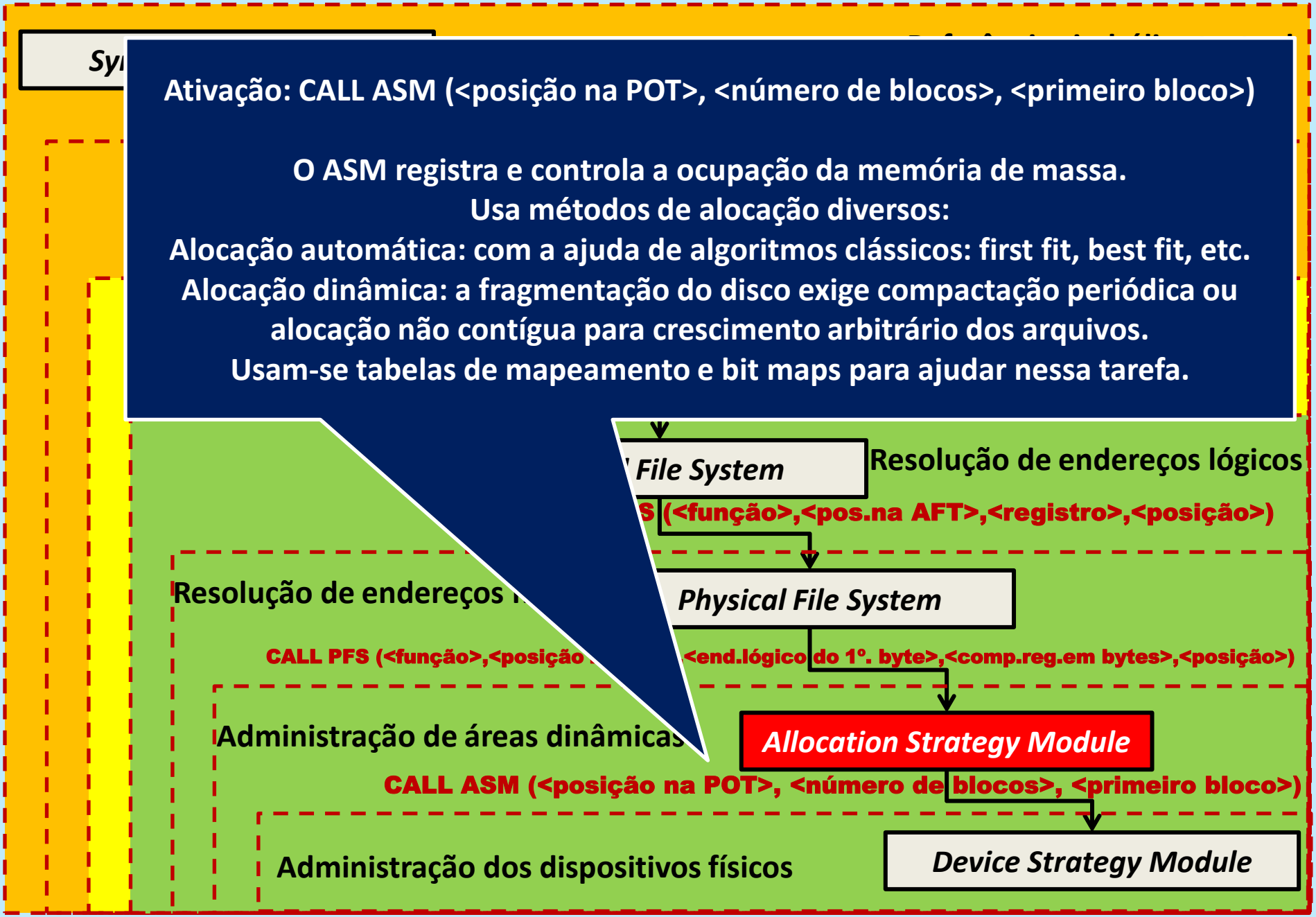
- Mantém informações acerca dos blocos não alocados (disponíveis) da memória de massa.
- É ativado pelo PFS através de um comando do tipo:

CALL ASM (<posição na POT>, <número de blocos>, <primeiro bloco>)

  - <posição na POT> e um número que indica Qual é a posição, na tabela POT, que contém as informações acerca da lei de mapeamento dos endereços para o periférico desejado.
  - <número de blocos> é a área de disco necessária à execução da transferência de dados que se deseja.
  - <primeiro bloco> indica a primeira posição (setor) escolhida na alocação dinâmica realizada pelo ASM, para onde a transferência de dados deve ser realizada.
- Pode ser ativado, alternativamente, pela rotina "CREATE" de inicialização de arquivos.

Ativação: CALL ASM (<posição na POT>, <número de blocos>, <primeiro bloco>)

O ASM registra e controla a ocupação da memória de massa.
Usa métodos de alocação diversos:
Alocação automática: com a ajuda de algoritmos clássicos: first fit, best fit, etc.
Alocação dinâmica: a fragmentação do disco exige compactação periódica ou alocação não contígua para crescimento arbitrário dos arquivos.
Usam-se tabelas de mapeamento e bit maps para ajudar nessa tarefa.



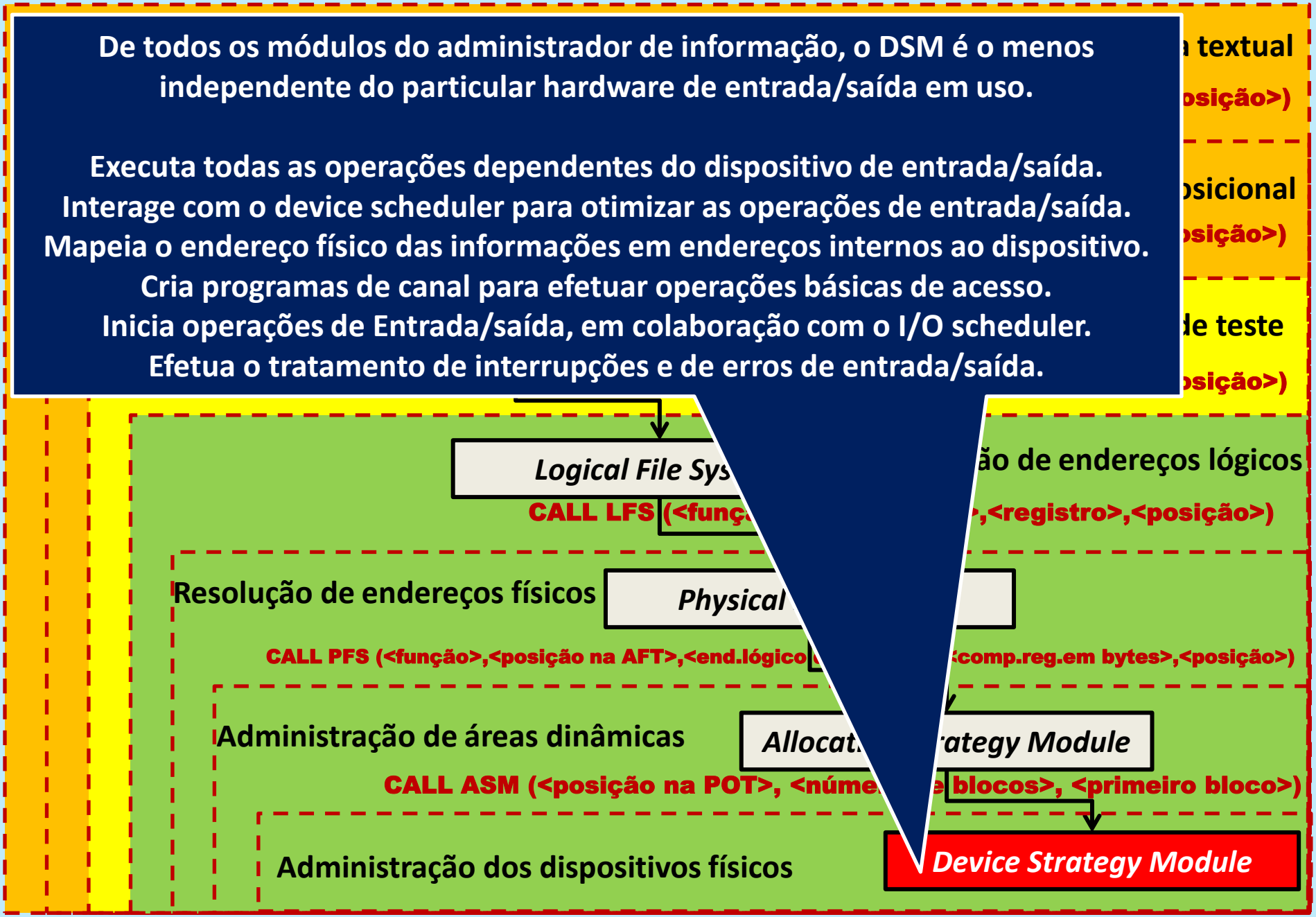
DSM – DEVICE STRATEGY MODULE

DSM (*Device Strategy Module*)

- Converte o número físico do bloco, fornecido pelo PFS, em um endereço que satisfaça as exigências de endereçamento do particular dispositivo.
- Executa **rotinas de entrada/saída** suficientes para iniciar as operações de leitura ou gravação solicitadas.
- Passa o controle ao *device scheduler*, para que a ordem de execução das entradas/saídas do sistema possam ser otimizadas.
- O DSM é, de todos os módulos que compõem o gerenciador de informações, **o menos independente da máquina**, já que está fisicamente vinculado aos dispositivos que deve gerenciar.

I/O Scheduler e *I/O Device Handler*

- Apesar de serem em geral considerados partes do administrador de informação, estes dois módulos são responsáveis pela entrada/saída física, razão pela qual, por questões didáticas, já foram estudados antes, como partes do gerenciador de dispositivos.

De todos os módulos do administrador de informação, o DSM é o menos independente do particular hardware de entrada/saída em uso.

Executa todas as operações dependentes do dispositivo de entrada/saída.
Interage com o device scheduler para otimizar as operações de entrada/saída.
Mapeia o endereço físico das informações em endereços internos ao dispositivo.
Cria programas de canal para efetuar operações básicas de acesso.
Inicia operações de Entrada/saída, em colaboração com o I/O scheduler.
Efetua o tratamento de interrupções e de erros de entrada/saída.