PCS 3446 - Sistemas Operacionais

Prof. João José Neto

AULA 04

Administração de Memória (3)

Memória Física Paginada

Administração de Memória Física Paginada

- O Conceito de **Paginação**.
  - **Blocos e páginas** – divisão da memória física e da memória lógica em porções de tamanho uniforme
  - **Mapeamento de endereços** – Alocação não contígua
- Alocação **Paginada Simples**.
  - **Programa completo alocado** na memória paginada
  - Uso da paginação somente como **técnica de alocação**
- **Proteção** de Páginas.
  - Proteção em **nível de página/bloco** apenas

Memória física e memória lógica

- O **hardware** oferece um espaço de endereços físicos, cujo conjunto constitui a **memória física**
- O **programa** ocupa conceitualmente um espaço de endereços próprio, dito **espaço de endereçamento lógico,** não obrigatoriamente alinhado ao espaço físico em que deverá ser alocado para sua execução
- Para que a memória física e a memória lógica possam ser **usadas de forma transparente**, é preciso efetuar um **mapeamento**, ou seja, uma conversão automática de endereços entre os dois espaços de endereçamento.

Compartilhamento de recursos

- Havendo recursos a serem utilizados por mais de um programa, surge a **concorrência** entre os programas pela posse desses recursos.
- Para que todos os programas possam utilizar os recursos disponíveis, torna-se necessário impor uma **disciplina** para a sua utilização.
- A **automatização da aplicação dessas regras** pode permitir que o **compartilhamento** dos recursos entre os programas seja feito de forma **transparente**.

Memória compartilhada

- Em um computador **multiprogramado**, a memória é um recurso que costuma ser compartilhado entre os programas que são nele executados.
- É possível simplesmente partilhar a memória física, através de uma **alocação simples**.
- Pode-se também permitir que, depois de alocada, uma parte da memória possa ser compartilhada, ou seja, **acessada concorrentemente por mais de um programa**, de acordo com alguma disciplina pré-estabelecida.

Bloco

- O **bloco** é a unidade básica de alocação das áreas de memória física.
- Em geral, refere-se a uma sequência de endereços físicos **contíguos**, tipicamente escolhida a partir de uma posição de memória cujo endereço seja múltiplo de alguma potência de dois.
- Em um bloco, geralmente o programa nele alocado tem **direitos de acesso** que variam de acordo com a natureza dos bytes que o preenchem: leitura apenas, leitura e escrita, execução apenas, etc.

Página

- Da mesma forma que a memória física se divide em blocos, o programa (lógico) deve também ser dividido em **páginas**, ou seja, em regiões de endereços lógicos contíguos, de **tamanho compatível com o tamanho dos blocos**, também iniciadas em endereços lógicos que sejam potências de dois.

Paginação da memória física

- Diz-se que um sistema adota a administração de ***memória física paginada*** quando o seu hardware implementa um **mapeamento** automático que viabiliza o esquema sugerido de divisão da memória em blocos, e dos programas em páginas, convertendo, **em tempo de execução,** os endereços lógicos nos correspondentes endereços físicos, para que o programa possa ser devidamente executado.

Alocação paginada simples

- Diz-se que o tipo da alocação é **paginada simples** quando o único método de alocação adotado é o da **paginação**.
- Esse esquema permite que **vários programas convivam na memória física**, viabilizando, através da técnica da **multiprogramação,** sua execução simultânea e o compartilhamento de recursos.
- A alocação paginada simples **exige que todo o programa esteja alocado na memória** para que possa ser executado, portanto **é uma técnica de alocação de memória física.**

Mapeamento de memória paginada

- Para que o programa possa ser devidamente executado, **os endereços lógicos devem ser convertidos dinamicamente em endereços físicos**.
- Isso é efetuado pelo **hardware**, através de uma operação automática de ***mapeamento de endereços***, que é feita com a ajuda de uma **tabela de mapeamento de páginas**.
- Para isso, o **endereço lógico** é interpretado como um par ordenado **(página, deslocamento)**, e o **endereço físico**, também como um par ordenado **(bloco, deslocamento)**.
- Como os tamanhos da página e do bloco são iguais, os deslocamentos coincidem, portanto **o mapeamento se reduz simplesmente a substituir o número da página pelo número do bloco** em que a página foi alocada.

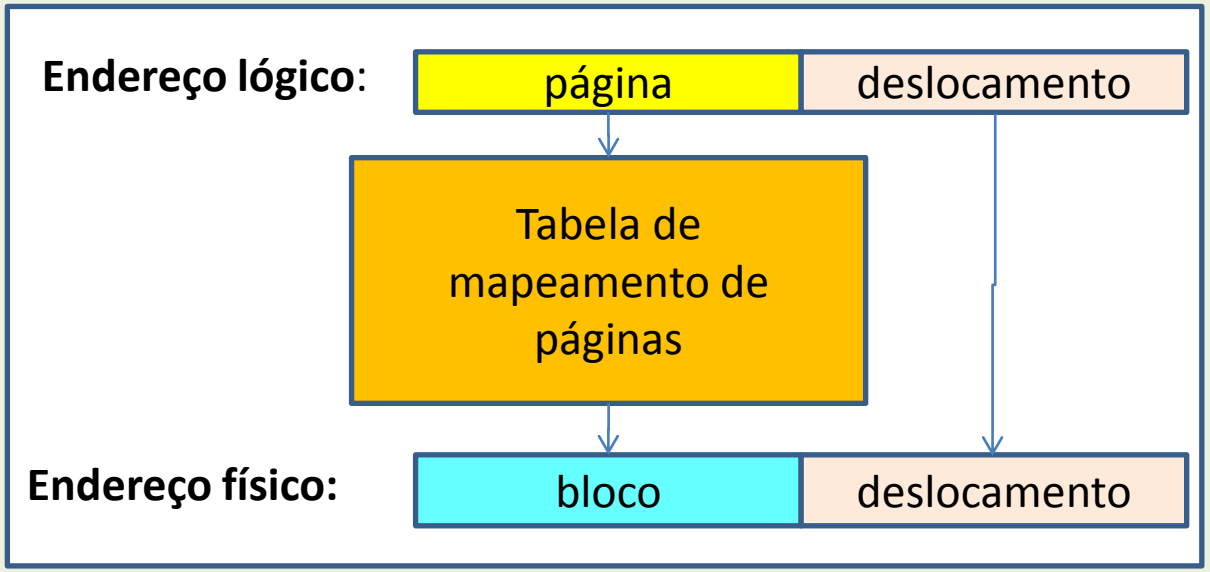
Page map

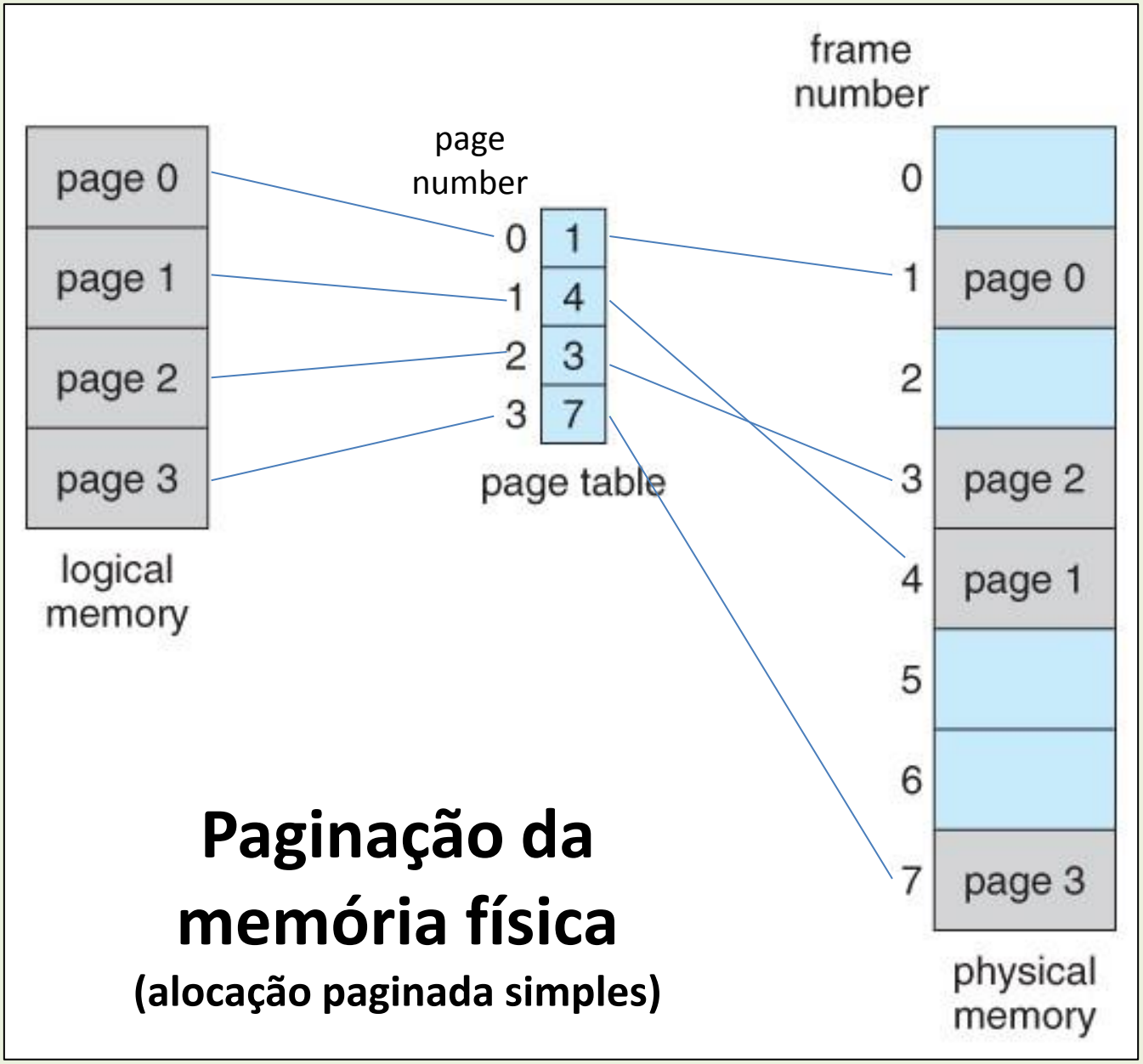
- Um **programa** pode então ser decomposto em **páginas**, sendo todas, exceto provavelmente a última, **do mesmo tamanho dos blocos** do hardware.
- Em sistemas operacionais com memória paginada, é necessário que **cada página seja associada ao bloco em que ela estiver alocada.**
- Essa associação deve ser usada pelo hardware para dinamicamente efetuar o mapeamento dos endereços referenciados pelo programa, **convertendo referências a páginas em referências a blocos**, e portanto gerando endereços físicos a partir dos endereços lógicos.

Page map table

- A **tabela de mapeamento de páginas** (uma para cada programa) deve ser **montada pelo *loader*** ao carregar o programa na memória, e deve ser preservada pelo sistema até que o programa termine.
- O ***processo de mapeamento*** é simples: indexa-se a tabela com o número da página, e o conteúdo de tal elemento da tabela é o número do bloco.

Processo de obtenção do endereço físico a partir do endereço lógico em memórias paginadas





Paginação da memória física

(alocação paginada simples)

Memory block table

- Para que o ***loader*** possa efetuar as alocações das páginas de um programa nos blocos de memória, é necessário que esteja informado acerca da **ocupação** dos mesmos por parte de **páginas já alocadas**, do mesmo programa ou de outros.
- Para isso, é conveniente manter um **mapa de ocupação da memória**, em que cada bloco existente seja representado por um código, cujo valor represente o estado de alocação do bloco (em branco = livre/ k = ocupado por uma página do programa k). Por exemplo:

	4	6	1	1	3	3			
	5	1		1	3	3	2		
6	5	1				2	2	7	7
6	5	1	1	4	3		4	7	7
6	5	4	3	3	3				7
			6	2	4	1			7
5	5					2		1	7
5	5							7	7

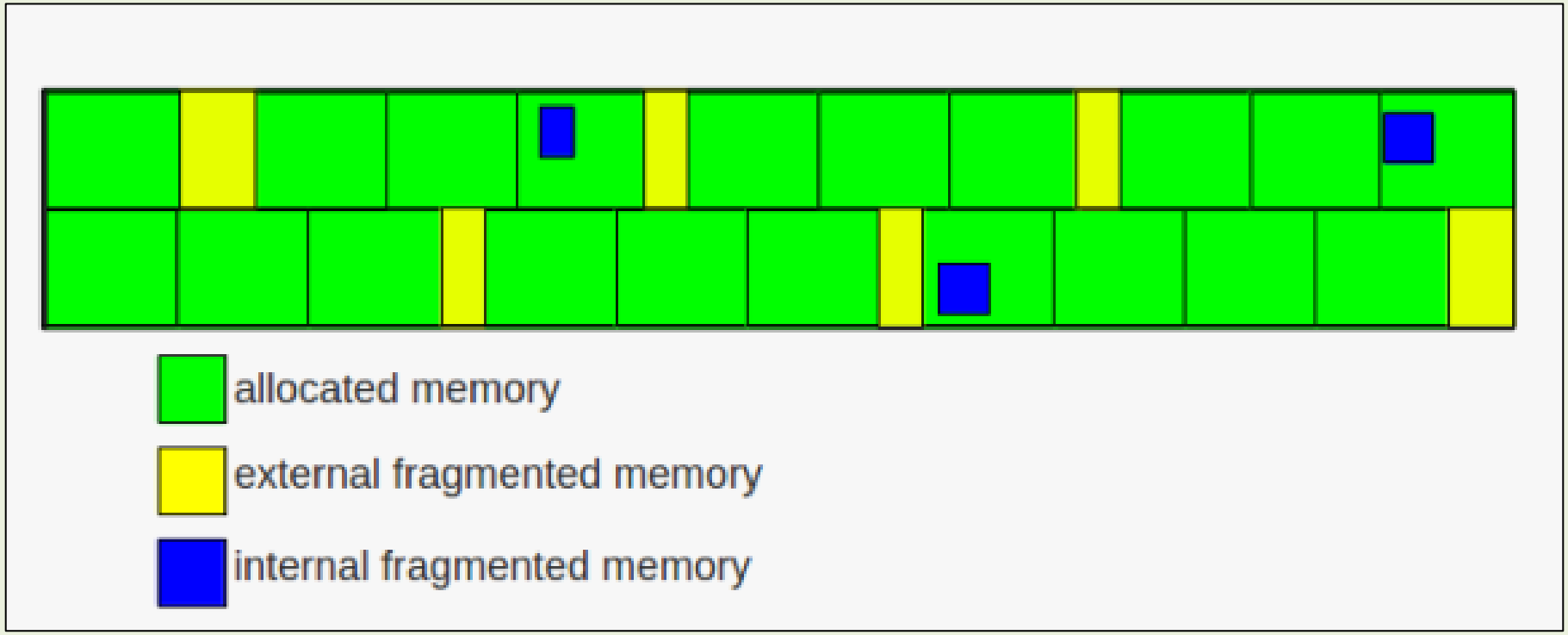
Memória associativa

- Em lugar de ser **indexada** por um endereço, com a finalidade de obter um conteúdo, uma ***memória associativa*** recebe **um valor**, e busca a célula que contém esse valor, fornecendo como saída o endereço dessa célula.
- Para servir ao propósito de acelerar as operações de busca na memória ou nas tabelas de mapeamento, é implementada **em hardware**, e neste caso, melhora muito a eficiência do processo de gerenciamento de memória.

Fragmentação interna

- No caso da administração de memória particionada, o problema da ***fragmentação*** era marcante, pois exigia o uso de algoritmos de ***compactação*** de memória.
- Com a **paginação**, foi **eliminado esse tipo de fragmentação**, porém cada programa passa a ter **em média meio bloco desperdiçado**, pois estatisticamente é muito improvável que a última página de cada programa tenha o comprimento exato para preencher totalmente todas as posições de memória do bloco.
- Este fenômeno é chamado ***fragmentação interna***.

Fragmentação: interna e externa

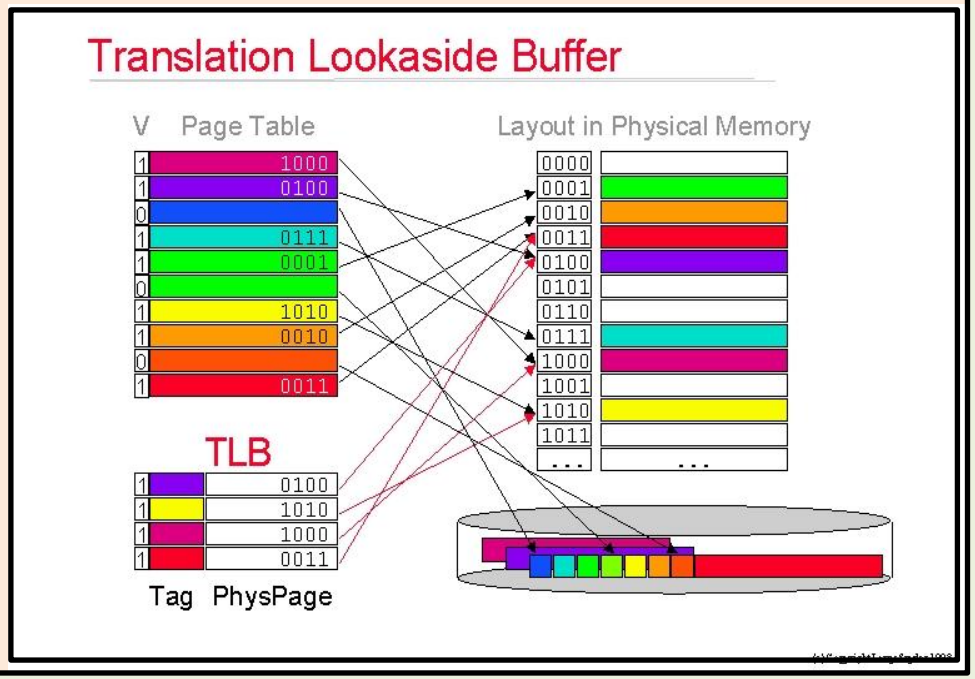


File map table

- O ***loader***, para carregar um programa na memória principal, deve provavelmente buscá-lo antes em um arquivo em disco.
- Para cada **arquivo** que contenha um **programa** em formato **executável**, a cada instância da execução de tal programa, deve ser reservada uma área em disco, pois durante sua execução costuma ser necessário **salvar o programa**, total ou parcialmente, devido a ocorrências no sistema ou no próprio programa.
- Uma **tabela de correspondência** entre as páginas do programa e os locais do disco em que são guardadas suas cópias constitui sua **tabela de mapeamento de arquivo** *(file map table)*.

Translation lookaside buffer

- Memória associativa de tamanho pequeno, usada para preservar e manter acessíveis ao sistema os endereços em que foram feitos os últimos acessos à memória.
- Disso pode resultar uma **economia de tempo**, pois evita recálculos desnecessários de endereços físicos associados a **acessos frequentes** a um pequeno conjunto de posições de memória, e nesse caso, a economia pode ser significativa.



Unallocated area table

- Muitas implementações de memórias paginadas mantêm uma tabela contendo os endereços de todos os **blocos não alocados**, às vezes indicando também o número de blocos contíguos disponíveis em cada local.
- Isso se destina a **facilitar a alocação** de um conjunto **de blocos consecutivos**, importante em determinadas situações para acelerar os processos de transferência de massas de dados.

Recuperação de memória para o sistema

- Uma das funções do sistema operacional é a de **recuperar,** para uso comum, os recursos alocados aos programas – neste caso**, a memória** – depois que tais programas deles não mais tiverem necessidade.
- No caso da memória isto pode acontecer **no decorrer da própria execução**, e não apenas ao seu término.
- Alguns gerenciadores de memória disponibilizam, por tal razão, serviços do sistema, destinados a promover a qualquer momento durante a execução do programa a **alocação e a recuperação** de áreas de memória.

Esquema *Key/Lock*

- Uma forma interessante de **proteção de memória,** que permite diversas variações, é a que utiliza, para cada programa ativo, um par de códigos, conhecidos como ***key-lock*** (chave-fechadura).
- Nesse esquema, o acesso só é permitido ao programa **quando esses dois códigos coincidirem**, e vetado em caso contrário.
- Tais códigos podem ser empregados inclusive para, em nível de página, **simbolizar** de maneira cifrada quais **tipos de acessos** podem ou não ser efetuados.

Prioridades de alocação

- Para determinadas aplicações, pode ser conveniente que se estabeleçam **prioridades** para a alocação de áreas de memória, dando-se preferência ao atendimento de certos programas, quer pela sua natureza de **urgência** (atendimento de **interrupções**, por exemplo) quer pela de **perigo detectado** (tais como ocorre em certas situações de **alarme,** nos sistemas de tempo real).

Tag

- Uma forma de marcação das páginas/blocos quanto ao **tipo de conteúdo** que contêm é usando bits de identificação (*tags*).
- A cada acesso à memória, os *tags* são consultados **por hardware**, comparados com bits de status que informam os direitos de acesso do programa.
- A execução então ocorre normalmente se houver **concordância** com as restrições vigentes.
- Em caso contrário, é gerada uma **interrupção** de **violação de direitos** de acesso.

Taxa de espera relativa de entrada/saída

- É uma das métricas usuais de avaliação da **eficiência do processamento**.
- Mede-se pela relação entre o tempo realmente utilizado em processamento pelo programa, e o tempo total que o programa permanece no sistema, desde o momento de sua chegada até o momento em que ele abandona o sistema.

Tempo efetivo de processador

- É o tempo total de uso real do processador pelo programa.
- Não são aqui contabilizados:
  - tempos de espera de entrada/saída,
  - overheads gastos pelo sistema operacional para efetuar processamentos em modo supervisor
  - tempos gastos pelo programa, nas filas do sistema operacional, à espera da conclusão de serviços solicitados

Overhead
de mapeamento de endereços

- No nosso caso, corresponde ao tempo adicional consumido pelo hardware e pelo software acionados pelos mecanismos propostos de mapeamento de endereços para a implementação da administração paginada de memória física.