República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) PI 0919075-9 B1

(22) Data do Depósito: 26/09/2009

(45) Data de Concessão: 24/12/2019

(54) Título: MÉTODO E SISTEMA DE COMPUTADOR PARA CONVERTER UMA PLURALIDADE DE PEQUENAS PÁGINAS ASSOCIADAS A UM OU MAIS PROCESSOS EM EXECUÇÃO

(51) Int.Cl.: G06F 12/02; G06F 12/06; G06F 12/08.

(30) Prioridade Unionista: 23/10/2008 US 12/257.091.

(73) Titular(es): MICROSOFT TECHNOLOGY LICENSING, LLC.

(72) Inventor(es): FORREST C. FOLTZ; DAVID N. CUTLER.

(86) Pedido PCT: PCT US2009058511 de 26/09/2009

(87) Publicação PCT: WO 2010/047918 de 29/04/2010

(85) Data do Início da Fase Nacional: 23/03/2011

(57) Resumo: MÉTODO E SISTEMA DE COMPUTADOR PARA CONVERTER UMA PLURALIDADE DE PEQUENAS PÁGINAS ASSOCIADAS A UM OU MAIS PROCESSOS EM EXECUÇÃO Tabelas de página no último nível de um sistema hierárquico de tabela de página são examinadas em relação a tabelas de página candidatas. Tabelas de página candidatas são convertidas em páginas grandes, tendo uma entrada de tabela de página em um nível antes do último nível do sistema hierárquico de tabela de página ajustado para ser associado com a página grande recentemente criada. Ao receber uma notificação de que uma página grande deve ser convertida em uma tabela de página, uma nova tabela de página é criada. Cada entrada na nova tabela de página é associada a um pequeno segmento de memória na página grande e uma entrada na tabela de página um nível antes do último nível em um sistema hierárquico de tabela de página é ajustado para ser associada a uma nova tabela de página.

Relatório Descritivo da Patente de Invenção para **"MÉTODO E SISTEMA DE COMPUTADOR PARA CONVERTER UMA PLURALIDADE DE PEQUENAS PÁGINAS ASSOCIADAS A UM OU MAIS PROCESSOS EM EXECUÇÃO".**

ANTECEDENTES

[0001] Processos executando em dispositivos de computação frequentemente requerem que os dados sejam usados em computações. Esses dados são armazenados tipicamente pelo sistema operacional na memória, tal como RAM. Essa memória é dividida em amostras denominadas páginas. Cada página é associada a um único endereço. Quando os processos requerem dados, os dados são referenciados pelo seu endereço único, e o endereço é usado para consultar a localização física da página para retornar os dados. Uma forma comum em que a conversão de endereço para localização física é realizada é mediante a ação de percorrer uma hierarquia de tabela de página. Tais hierarquias compensam o tamanho das páginas que são endereçadas com o número de níveis na hierarquia. Contudo, o tamanho das páginas também determina quão eficientemente o espaço de memória é usado, com as páginas maiores sendo menos eficientes. Portanto, há um equilíbrio direto entre eficiência de espaço (devido ao tamanho de página) e eficiência de tempo de conversão (devido ao número de páginas na hierarquia de tabela de página).

[0002] Um fator adicional na determinação da eficiência de um sistema de tabela de página consiste nas necessidades dos processos. Se os processos requerem tipicamente grandes quantidades de dados, então páginas maiores podem na realidade ser eficientes em termos de utilização de memória. Contudo, se os processos exigem tipicamente pequenas quantidades de dados, então páginas menores serão mais eficientes. Uma vez que os processos de ambos os tipos tendem a operar em dispositivos de computação, um método de suportar

dinamicamente ambos levaria à maior eficiência. Suporte de sistema operacional para páginas grandes também não é tão robusto nos dispositivos de computação como o suporte para páginas de tamanho menor. Isso leva a um desafio adicional no uso de páginas grandes.

SUMÁRIO

[0003] Esse sumário é provido para introduzir uma seleção de conceitos em uma forma simplificada que são descritos adicionalmente abaixo na Descrição Detalhada. Esse sumário não pretende identificar características essenciais ou características fundamentais da matéria em estudo reivindicada, nem pretende ser usado como um meio auxiliar na determinação do escopo da matéria em estudo reivindicada.

[0004] Modalidades da presente invenção se referem à varredura do último nível em uma hierarquia de tabela de página para localizar entradas de tabela de página (PTEs) candidatas para conversão em mapeamentos de páginas grandes. Quando as PTEs candidatas são localizadas, essas PTEs candidatas são convertidas em páginas grandes mediante localização de um segmento grande, contíguo de memória física, transferência dos dados associados a todas as PTEs na página de tabela de página candidata para o segmento localizado de memória, e então ajustar uma PTE em uma página de tabela de página um nível antes do último nível da hierarquia de tabela de página a ser associada com a página grande recentemente criada. Em algumas modalidades, quando uma notificação é recebida, indicando uma página grande que deve ser convertida de volta em páginas pequenas, uma nova página de tabela de página é criada. Cada PTE na nova página de tabela de página é associada a um pequeno segmento da página grande e uma PTE na tabela de página um nível antes do último nível do sistema de tabela de página, hierárquico é ajustado para ser associado com a nova página de tabela de página.

BREVE DESCRIÇÃO DOS DESENHOS

[0005] A presente invenção é descrita em detalhe abaixo com referência às figuras de desenho anexo, em que:

[0006] A Figura 1 ilustra um diagrama de blocos de um dispositivo de computação exemplar adequado para uso na implementação da presente invenção;

[0007] A Figura 2 é um diagrama de um leiaute de memória física típica conforme usado pelos sistemas operacionais e processos de usuário;

[0008] A Figura 3 ilustra uma relação exemplar entre uma tabela de página e memória física;

[0009] A Figura 4 ilustra um sistema exemplar de tabela de página hierárquica;

[00010] A Figura 5 é um diagrama de fluxo mostrando um método para encontrar tabelas de páginas candidatas para conversão em páginas grandes e realizar a conversão;

[00011] A Figura 6 é um diagrama de fluxo mostrando um método para receber uma notificação de que uma página grande deve ser convertida em uma tabela de página associada com páginas pequenas e realizar a conversão; e

[00012] A Figura 7 é um diagrama de fluxo mostrando um método para receber uma notificação de que uma página grande deve ser convertida em uma tabela de página associada com páginas pequenas, ou receber uma determinação de interrupção indicando que está na hora de varredura no sentido de tabelas de páginas candidatas para conversão em páginas grandes.

DESCRIÇÃO DETALHADA

[00013] A matéria em estudo da presente invenção é descrita aqui com especificidade para satisfazer às exigências legais. Contudo, a própria descrição não pretende limitar o escopo dessa patente. Mais propriamente, os inventores consideraram que a matéria em estudo

reivindicada também poderia ser incorporada de outras formas, para incluir diferentes etapas ou combinações de etapas similares àquelas descritas nesse documento, em conjunto com outras tecnologias presentes ou futuras. Além disso, embora os termos, "etapa" e/ou "bloco" possam ser usados aqui para conotar diferentes elementos de métodos empregados, os termos não devem ser interpretados como significando qualquer ordem específica entre ou dentre as várias etapas aqui reveladas a menos que, e exceto quando a ordem das etapas individuais for descrita explicitamente.

[00014] Modalidades da presente invenção são dirigidas para localização de forma oportunista de grupos de PTEs que poderiam ser convertidos em página grande e realização de uma conversão. Adicionalmente, quando uma página de tabela de página tiver sido convertida em uma página grande, o processo inverso pode ser realizado em reação a uma notificação a partir do sistema operacional.

[00015] De acordo com algumas modalidades da presente invenção, o subsistema de memória de um dispositivo de computação gerencia um recurso de memória compartilhada. Os dados requeridos para computação por um ou mais processos são armazenados no recurso de memória compartilhada. Tipicamente, processos executando no dispositivo de computação não têm conhecimento da localização física dos dados. Em vez disso, esses processos são apresentados com um espaço de endereço mapeando endereços para localizações físicas na memória. O um ou mais processos executando no dispositivo de computação utilizam o endereço para se referir aos dados exigidos para computação. O subsistema de memória do dispositivo de computação lida com a conversão a partir de endereço para localização física, realizando consultas de endereço.

[00016] Nos modernos dispositivos de computação a memória física é dividida em segmentos referidos como páginas. Essas páginas re-

presentam o tamanho mínimo de dados que pode ser representado pela hierarquia de tabela de página. Tabelas de página são usadas pelo subsistema de memória do dispositivo de computação para mapear os endereços virtuais para localizações físicas na memória. Há alguns leiautes possíveis para sistemas de tabela de página; contudo, os mapeamentos mais comuns a partir de endereços para localizações de memória física utilizam múltiplas consultas de tabela de página hierárquica que são descritas em detalhe abaixo. Essas hierarquias permitem que tamanhos fixos de endereço (tipicamente medidos em bits) se refiram a quantidades grandes de memória física. Tais consultas de tabela hierárquica requerem múltiplos acessos de memória para localizar uma página física associada a um determinado endereço virtual. Quanto maior forem os níveis no sistema de tabela de página hierárquica, mais dispendiosas são as operações de acesso aos dados em termos de tempo para a conversão de endereço para memória física. Contudo, também há um equilíbrio entre o número de níveis na hierarquia de tabela de página e o tamanho de página. Número menor de níveis na hierarquia de tabela de páginas significa tamanho de página maior. Portanto, para aplicações utilizando pequenos segmentos de dados, pequenos tamanhos de página e, portanto, hierarquias mais profundas permitem menos desperdício de memória. Contudo, para aplicações utilizando uma grande quantidade de dados, tamanhos maiores de página reduzirão o número de consultas de tabela de página exigidas para localizar os dados exigidos e, portanto, aumentam a eficiência de consulta.

[00017] Quando uma peça de dados específica não mais é necessária ou não foi acessada por um período de tempo limite, é comum que os subsistemas de memória salvem aquela peça de dados no disco, liberando memória para os dados que são mais frequentemente ou atualmente necessários. Esse processo é denominado permuta de

memória. Contudo, muitos subsistemas de memória podem apenas permutar certo tamanho fixo de página. Portanto, qualquer mecanismo que crie páginas maiores do que esse tamanho fixo teria que ter a capacidade de fracionar as páginas grandes em múltiplas páginas de tamanho menor no caso de alguma parte da página grande ser esvaziada. Há muitas outras situações adicionais em que uma página grande precisaria ser dividida em páginas de tamanhos menores por intermédio de um subsistema de memória.

[00018] Consequentemente, uma modalidade da invenção se refere aos meios de armazenamento legíveis por computador incorporando instruções utilizáveis por computador para realizar um método de converter uma pluralidade de pequenas páginas associadas a um ou mais processos operando em um dispositivo de computação em uma página grande. Cada uma das páginas é associada a uma entrada em uma tabela de páginas a partir de um sistema de tabela de página, hierárquico contendo ao menos dois níveis de tabelas de página. O método inclui examinar o último nível do sistema de tabela de página, hierárquico em termos de PTEs candidatas, que são tabelas de página com ao menos um limite de entradas associado com as páginas. O método então localiza um segmento de memória fisicamente contíguo suficientemente grande para armazenar cada uma das páginas associadas com as entradas na tabela de página candidata e copia os segmentos de memória em cada uma das páginas a serem localizadas no segmento de memória. O método ajusta uma entrada de tabela de página em uma tabela de página um nível antes do último nível no sistema de tabela de página, hierárquica a ser associado com a página grande recentemente criada.

[00019] De acordo com outras modalidades, a invenção se refere aos meios legíveis por computador armazenando instruções executáveis por computador incorporando um método de converter uma pági-

na grande em uma pluralidade de pequenas páginas associadas com um ou mais processos executando em um sistema de computador. Cada uma das páginas é associada com uma entrada de uma tabela de página em um sistema hierárquico de tabela de página. O método inclui receber uma notificação de sistema operacional indicando que uma página grande deve ser convertida em um grupo de pequenas páginas. Ao receber a notificação, uma nova tabela de página é criada e as entradas na nova tabela de página são associadas com pequenos segmentos da página grande. O método inclui ajustar uma entrada a partir de uma tabela de página um nível antes do último nível do sistema hierárquico de tabela de página a ser associado com a nova tabela de página.

[00020] De acordo com uma modalidade adicional, a invenção se refere aos meios legíveis por computador armazenando instruções executáveis por computador incorporando um método de examinar um último nível de um sistema hierárquico de tabela de página, contendo ao menos dois níveis de tabelas de página, em cada um de vários espaços de endereço mencionados com um ou mais processos executando em um sistema de computação. Essa varredura envolve tentar identificar as tabelas de página candidata, que são tabelas de página para as quais as entradas são associadas a um ou mais segmentos de memória física. O método inclui ainda localizar um segmento de memória composto de segmentos contíguos de memória física suficientemente grande para armazenar cada um dos vários segmentos de memória física associados a todas as entradas em uma tabela de página candidata e copiar esses segmentos de memória física no segmento de memória recentemente carregado. O método libera o segmento de memória contendo a tabela de página candidata e ajusta uma entrada de tabela de página em uma tabela de página um nível antes do último nível no sistema hierárquico de tabela de página que

foi associado com a tabela de página candidata a ser associada com o segmento recentemente localizado da memória, denominado página grande. O método inclui ainda receber uma indicação a partir de um subsistema de memória incapaz de permutar páginas que indicam que um ou mais segmentos de uma página grande devem ser permutados. O método inclui ainda criar uma nova tabela de página, com cada entrada na nova tabela de página sendo associada a um segmento da página grande contendo o segmento ou segmentos que devem ser permutados. O método inclui ainda ajustar uma entrada de tabela de página um nível antes do último nível do sistema hierárquico de tabela de página que foi previamente associado com a página grande a ser associada com a nova tabela de página.

[00021] Tendo descrito resumidamente uma visão geral das modalidades da presente invenção, um ambiente de operação exemplar no qual as modalidades da presente invenção podem ser implementadas é descrito abaixo para prover um contexto geral para vários aspectos da presente invenção. Com referência inicialmente à Figura 1, em particular, um ambiente de operação exemplar para implementar as modalidades da presente invenção é mostrado e designado geralmente como o dispositivo de computação 100. O dispositivo de computação 100 é apenas um exemplo de um ambiente de computação adequado e não pretende sugerir qualquer limitação em relação ao escopo de uso ou funcionalidade da invenção. O dispositivo de computação 100 também não deve ser interpretado como tendo qualquer dependência ou exigência relacionada a qualquer um componente ou combinação de componentes ilustrados.

[00022] A invenção pode ser descrita no contexto geral de código de computador ou instruções utilizáveis por máquinas, incluindo instruções executáveis por computador tais como módulos de programa, sendo executados por um computador ou outra máquina, tal como um

assistente pessoal de dados ou outro dispositivo de mão. Geralmente, módulos de programa incluindo rotinas, programas, objetos, componentes, estruturas de dados, etc., se referem ao código que realiza tarefas específicas ou implementam tipos de dados abstratos, específicos. A invenção pode ser praticada em uma variedade de configurações de sistema, incluindo dispositivos de mão, meios eletrônicos de consumidor, computadores de uso geral, dispositivos de computação mais especializados, etc. A invenção também pode ser praticada em ambientes de computação distribuída onde as tarefas são realizadas por dispositivos de processamento remoto que são ligados através de uma rede de comunicação.

[00023] Com referência à Figura 1, o dispositivo de computação 100 inclui um barramento 110 que acopla direta ou indiretamente os seguintes dispositivos: memória 112, um ou mais processadores 114, um ou mais componentes de armazenamento externo 116, portas de entrada/saída (E/S) 118, componentes de entrada 120, componentes de saída 121, e um fornecimento de energia, ilustrativo 122. O barramento 110 representa o que pode ser um ou mais barramentos (tal como um barramento de endereço, barramento de dados, ou combinação dos mesmos). Embora os vários blocos da Figura 1 sejam mostrados com linhas com a finalidade de clareza, na realidade, delinear os vários componentes não é muito claro, e metaforicamente, as linhas seriam mais exatamente cinzas e confusas. Por exemplo, muitos processadores têm memórias. Reconhecemos que tal é a natureza da técnica, e reiteramos que o diagrama da Figura 1 é meramente ilustrativo de um dispositivo de computação exemplar que pode ser usado em conexão com uma ou mais modalidades da presente invenção. Faz-se agora distinção entre tais categorias como “estação de trabalho”, “servidor”, “laptop”, “dispositivo de mão”, etc., uma vez que todos são considerados dentro do escopo da Figura 1 e referência ao “dis-

positivo de computação".

[00024] O dispositivo de computação 100 inclui tipicamente uma variedade de meios legíveis por computador. Os meios legíveis por computador podem ser quaisquer meios disponíveis que podem ser acessados pelo dispositivo de computação 100 e inclui meios voláteis e não voláteis; meios removíveis e não removíveis. Como exemplo, e não como limitação, meios legíveis por computador podem compreender meios de armazenamento de computador e meios de comunicação. Meios de armazenamento de computador incluem meios voláteis, e não voláteis; removíveis e não removíveis implementados em qualquer método ou tecnologia para armazenamento de informação tal como instruções legíveis por computador, estruturas de dados, módulos de programa ou outros dados. Meios de armazenamento de computador incluem, mas não são limitados a RAM, ROM, EEPROM, memória flash ou outra tecnologia de memória, CD-ROM, discos versáteis digitais (DVD), ou outro meio de disco ótico, cassetes magnéticos, fita magnética, meio de armazenamento de disco magnético ou outros dispositivos de armazenamento magnético, ou qualquer outro meio que possa ser usado para armazenar a informação desejada e que possa ser acessado pelo dispositivo de computação 100.

[00025] A memória 112 inclui meios de armazenamento de computador na forma de memória volátil. Dispositivos de hardware exemplares incluem memória de estado sólido, tal como RAM. Meio de armazenamento externo 116 inclui meios de armazenamento de computador na forma de memória não volátil. A memória pode ser removível, não removível, ou uma combinação das mesmas. Dispositivos de hardware exemplares incluem memória de estado sólido, unidades de disco rígido, unidades de disco ótico, etc. O dispositivo de computação 100 inclui um ou mais processadores que lêem os dados a partir das várias entidades tal como memória 112, meio de armazenamento ex-

terno 116 ou componentes de entrada 120. Os componentes de saída 121 apresentam indicações de dados a um usuário ou outro dispositivo. Componentes de saída exemplares incluem um dispositivo de exibição, altofalante, componente de impressão, componente de vibração, etc.

[00026] Portas E/S 118 permitem que o dispositivo de computação 100 seja acoplado logicamente a outros dispositivos incluindo componentes de entrada 120 e componentes de saída 121, alguns dos quais podem ser embutidos. Componentes ilustrativos incluem um microfone, joystick, elemento de jogos, antena de prato de satélite, escâner, impressora, dispositivo sem fio, etc.

[00027] De acordo com uma modalidade da invenção, o dispositivo de computação 100 poderia ser usado como um hipervisor, que é uma plataforma de virtualização que abstrai os componentes físicos do dispositivo de computação 100, tais como os componentes de entrada 120 e a memória 112 a partir do sistema operacional ou sistema executando no dispositivo de computação 100. Tais hipervisores permitem que múltiplos sistemas operacionais executem em um único dispositivo de computação 100 através de tal abstração, permitindo que cada sistema operacional independente tenha acesso à sua própria máquina virtual. Em dispositivos de computação de hipervisor, o custo indireto associado com a ação de percorrer hierarquias de tabela de página é ainda maior e os benefícios de utilizar páginas grandes são ainda maiores do que nos sistemas executando sistemas operacionais únicos que têm acesso direto aos componentes do dispositivo de computação 100.

[00028] Voltando-se para a Figura 2, uma memória física 200, tal como uma RAM, é dividida em um número de seções. De acordo com algumas modalidades da invenção, a memória é dividida em duas partições principais, um espaço de memória de sistema operacional 201 e

um espaço de memória de usuário 202. Um subsistema de memória de um sistema operacional executando no dispositivo de computação gerencia a memória física 200, permitindo que aplicações de usuário utilizem porções do espaço de memória de usuário 202. As aplicações, contudo, podem não ter acesso aos locais contíguos de memória. Com referência à Figura 2, de acordo com algumas modalidades da presente invenção, o espaço de memória 202 é dividido em páginas (representadas por caixas), que são distribuídas entre duas aplicações hipotéticas apenas para fins de ilustração e não de limitação; espaço de aplicação 1 é representado por x's (por exemplo, segmento de memória 203) e espaço de aplicação 2 é representado por /'s (por exemplo, segmento de memória 204). Páginas de memória livre são evidentes no diagrama (por exemplo, segmento de memória 205). O espaço de memória de sistema operacional 201 pode ser usado para um número de finalidades, uma das quais é de armazenar as tabelas de páginas 206 que contém o mapeamento a partir do espaço de endereço para a memória física. Para as aplicações, esses mapeamentos associam as páginas no espaço de memória de usuário 202 onde os dados são armazenados com endereços.

[00029] Conforme mostrado na Figura 3, de acordo com uma modalidade da presente invenção, uma tabela de página 301 inclui entrada 303, cada uma das quais é associada a uma página específica no espaço de memória de usuário, armazenadas na memória física 304. Observar que as entradas 303 na tabela de página 301 podem não ser necessariamente associadas com as páginas contíguas 304 na memória física.

[00030] Com referência agora à Figura 4, de acordo com as várias modalidades da presente invenção, os endereços 401 são representados por sequências de bits. Esses endereços são mapeados através de um sistema hierárquico de tabela de página 402. Como exemplo e

não como limitação, considere um esquema de endereçamento de 48 bits 401 e um sistema de tabela de página hierárquica de quatro níveis 402. O endereço de 48 bits 401 é dividido em cinco seções. Os primeiros nove bits 403 são usados para indexar em uma primeira tabela de página 402. A entrada localizada na primeira tabela de página 404 pelos primeiros nove bits 403 do endereço 401 é associada a um segmento de memória 422 armazenando uma segunda tabela de página 406. Os segundos nove bits 405 são indexados na segunda tabela de página 406. A entrada localizada na segunda tabela de página 406 é associada a um segmento de memória 422 contendo uma terceira tabela de página 408. Os terceiros nove bits 407 do endereço 401 indexam na terceira tabela de página 408. A entrada localizada na terceira tabela de página 408 é associada a um segmento de memória 423 contendo uma quarta tabela de página 410. O quarto nove bits 409 do endereço 401 indexam na quarta tabela de página 410. A entrada localizada na quarta tabela de página 410 é associada a um segmento de memória 424 na memória de espaço de usuário contendo uma página 412. Os últimos 12 bits 411 do endereço 401 indexam na página 412. O segmento de memória na página 412 no índice dado pelos últimos 12 bits 412 é representado pelos dados referidos pelo endereço 401. Conforme pode ser visto há pelo menos um acesso de memória por consulta de tabela de página no processo de consultar os dados endereçados por intermédio de um sistema hierárquico de tabela de página.

[00031] Aqueles versados na técnica reconhecerão que os tamanhos de endereço, específicos, o número de tabelas de página, o número de níveis no sistema hierárquico de tabela de páginas, e o tamanho das páginas podem ser variados. Apenas como exemplo e não como limitação, os tamanhos de página podem ser de 4KB, 2MB ou 1GB. Tamanhos de endereço podem variar, por exemplo, de 32 bits a

64 bits. Dado o exemplo na Figura 4, cada tabela de página tem 512 entradas (2^9) e cada página tem 4KB (12^{12}). São realizadas quatro consultas de tabela de página para localizar os dados em uma página. Se todos os dados associados com todas as 512 entradas em uma tabela de páginas devessem ser combinados em uma única página, a página resultante (denominada página grande) seria de 2MB e requereria apenas três consultas de tabela de página no sistema hierárquico de tabela de página para localização.

[00032] Voltando-se para a Figura 5, é provido um diagrama de fluxo que ilustra um método 500 para encontrar uma tabela de página candidata para converter em uma página grande e assim converter a tabela de página (bloco 550 contém a etapas do método sem a porção de interrupção mostrada no bloco 503, tudo discutido abaixo). Mostrado no bloco 501, o último nível do sistema hierárquico de tabela de página é explorado para tabelas de página candidatas para conversão em páginas grandes. Por exemplo, o último nível da hierarquia de tabela de página da Figura 4, na qual existe a quarta tabela de página 410, poderia ser explorada em relação a tabelas de página candidatas. Aqueles versados na técnica reconhecerão que uma ampla variedade de critérios poderia ser usada para determinar se uma tabela de página é uma candidata para conversão em uma página grande. Apenas como exemplo e não como limitação, tais critérios poderiam incluir a descoberta de uma tabela de página completa ou a descoberta de uma tabela de páginas com um limite de entradas completas. Uma tabela de página completa é aquela em que todas as entradas da tabela de página são associadas às localizações na memória física. De acordo com uma modalidade da invenção, tal varredura envolve examinar através de cada uma das tabelas de páginas associadas com as entradas nas tabelas de página um nível antes do último nível e examinar as tabelas de página do último nível, encontradas, para verificar se

elas constituem uma tabela de página completa. Aqueles versados na técnica reconhecerão que há muitas formas nas quais um limite poderia ser definido, incluindo, mas não limitado a uma percentagem das entradas sendo associadas com locais de memória física ou um número total de entradas associadas aos locais de memória física.

[00033] Mediante varredura do último nível do sistema hierárquico de tabela de página (por exemplo, o nível no qual a tabela de página 410 está localizada na Figura 4) uma ou mais tabelas de página candidatas poderiam ser identificadas (vide bloco 502). Se nenhuma tabela de página candidata tiver sido identificada, então há um retardo de tempo 503 antes de outra varredura ser realizada no bloco 501. Esse retardo de tempo 503 é um parâmetro que poderia ser ajustado por um programador, administrador de sistema, usuário, ou alguém mais com acesso apropriado ao sistema. Se, contudo, uma tabela de página candidata tiver sido identificada, então um segmento de memória contígua suficientemente grande para armazenar os dados associados com cada entrada na tabela de página candidata é localizado, como mostrado no bloco 504.

[00034] Em modalidades, localizar um segmento de memória envolve examinar a memória física para um número suficiente de segmentos contíguos de memória para armazenar todas as entradas associadas com a tabela de página candidata. Lembrar que uma tabela de página pode não ter entradas contíguas que são associadas com segmentos contíguos de memória física. Contudo, quando as entradas na tabela de página candidata são convertidas em uma página grande, elas devem ser armazenadas na ordem das entradas na tabela de página com a qual elas são associadas. De acordo com uma modalidade da invenção, a localização de um segmento de memória é simplesmente uma questão de examinar a memória física e encontrar um segmento contíguo grande de memória (por exemplo, 2MB). Em algu-

mas modalidades, essa varredura pode ser realizada mediante varredura de um banco de dados de número de quadro de página contendo o estado de todas as páginas físicas no sistema. Adicionalmente, o segmento contíguo grande de memória poderia ser restrito a começar em um limite de byte predeterminado. Como exemplo, e não como limitação, considerando o exemplo acima utilizando 512 páginas de tamanho pequeno de 4KB para combinar em uma página grande de 2MB, o limite de bytes predeterminados poderia ser um limite de byte de 2MB. Aqueles versados na técnica reconhecerão que muitos outros valores para o limite de byte predeterminado poderiam ser usados. De acordo com outra modalidade da invenção, se segmentos contíguos não suficientes de memória puderem ser encontrados, então uma subrotina de gerenciamento de memória é ativada que cria ativamente um segmento contíguo grande de memória mediante deslocamento dos dados armazenados para segmentos livres afastados de um local específico na memória, e ajustando suas respectivas entradas de tabela de página. Desse modo um segmento contíguo grande de memória é criado para uso na conversão de tabela de página grande.

[00035] Quando um segmento contíguo de memória de tamanho suficiente tiver sido localizado ou criado, todos os segmentos físicos de memória associados com as entradas na tabela de página candidata são copiados para o segmento de memória localizado, conforme mostrado no bloco 505. Em uma modalidade da presente invenção, quando os segmentos físicos de memória são copiados para o segmento localizado, a localização original da memória física é liberada. Em outra modalidade da invenção, as localizações de memória originais de cada um dos segmentos de memória associados com cada entrada da tabela de página candidata também mantém suas cópias dos dados.

[00036] Conforme mostrado no bloco 506, entrada de tabela de pá-

gina um nível antes do último nível do sistema hierárquico de tabela de página (por exemplo, tabela de página 408 da Figura 4) é associado com a nova página grande. Em uma modalidade da invenção, a tabela de página convertida é liberada e a entrada de tabela de página a partir de um nível antes do último nível no sistema hierárquico de tabela de página que foi associada com a tabela de página liberada é ajustado para ser associado com a nova página grande. Após converter candidatos em páginas grandes, há um retardo de tempo no bloco 503 antes de outra varredura em relação a novas tabelas de página candidata ser iniciada. Esse retardo de tempo no bloco 503 é um parâmetro que poderia ser ajustado por um programador, administrador de sistema, usuário, ou qualquer outro com acesso apropriado ao sistema.

[00037] Voltando-se para a Figura 6, é provido um diagrama de fluxo que ilustra um método 600 para converter uma página grande em entradas de tabela de página associadas com múltiplas páginas de tamanho menor. De acordo com uma modalidade da presente invenção, uma notificação de sistema operacional é recebida no bloco 601 identificando uma página grande a ser convertida em páginas pequenas. Aqueles versados na técnica reconhecerão que há muitos eventos que poderiam ativar tal notificação. Apenas como exemplo, e não como limitação, tais eventos incluem um segmento da página grande sendo programado para permuta para disco em um sistema com um sistema operacional incapaz de permutar páginas grandes e as entradas de tabela de página associadas com memória pertencendo a um espaço de memória de aplicação que está sendo destruído.

[00038] Ao receber uma notificação indicando uma página grande a ser convertida, uma nova tabela de página é criada como mostrado no bloco 602. De acordo com uma modalidade da invenção, essa criação envolve alocar memória no espaço de memória de sistema operacional para uma nova tabela. Quando a tabela de página é criada, cada

entrada na nova tabela de página é associada a um segmento de tamanho menor da página grande no bloco 603, até que todos os segmentos da página grande sejam associados com alguma entrada na nova tabela de página. Continuando com o exemplo da Figura 4, cada uma das 512 entradas de tabela de página na tabela de página nova seria associada a um segmento de 4KB da página grande.

[00039] Finalmente, uma entrada de tabela de página a partir de um nível acima do último nível do sistema hierárquico de tabela de página (por exemplo, o nível no qual a tabela de página 408 é alocada na Figura 4) é ajustada para ser associada com a nova tabela de página, conforme mostrado no bloco 604. De acordo com uma modalidade da invenção, a entrada a partir da tabela de página um nível antes do último nível do sistema hierárquico de tabela de página associado com a nova tabela de página foi a entrada previamente associada com a página grande.

[00040] De acordo com uma modalidade adicional, a Figura 7 apresenta um método 700 de converter uma tabela de página em uma página grande e converter páginas grandes em tabelas de páginas associadas com diversas páginas de tamanho menor. Em primeiro lugar o método envolve esperar por um evento, conforme mostrado no bloco 701. Como exemplo, e não como limitação, o evento poderia ser uma interrupção ou uma notificação de sistema operacional. Aqueles versados na técnica reconheceriam que há diversos outros exemplos que poderiam ativar qualquer um dos tipos de conversão. Quando ocorre um evento, uma decisão é tomada. Se o evento foi uma interrupção indicando que um retardo de tempo expirou 702, é feita uma tentativa de converter uma tabela de página a partir do último nível de um sistema hierárquico de tabela de página em uma página grande, por exemplo, de acordo com o método 550 da Figura 5. Esse retardo de tempo é um parâmetro que poderia ser ajustado por um programador,

administrador de sistema, usuário, ou alguém mais com acesso apropriado ao sistema. Se o evento for uma notificação de sistema operacional 702, então uma página grande é convertida em uma tabela de página de entradas indicando páginas de tamanho menor, por exemplo, de acordo com o método 600 da Figura 6. A partir da conclusão de qualquer um de método de tentar converter uma tabela de página em uma página grande, ou método de converter uma página grande em uma tabela de página com entradas associadas a várias páginas de tamanho menor, um período de espera é outra vez introduzido no bloco 701. Esse período de espera outra vez expira quer seja na chegada de outra notificação de sistema operacional ou a expiração de um retardo de tempo.

[00041] Muitos arranjos diferentes dos vários componentes ilustrados, assim como componentes não mostrados, são possíveis sem se afastar do espírito e escopo da presente invenção. Modalidades da presente invenção foram descritas com o propósito de ilustração mais propriamente do que de restrição. Modalidades alternativas se tornarão evidentes para aqueles versados da técnica as quais não se afastam desse escopo. Aqueles versados na técnica podem desenvolver meios alternativos de implementar os aperfeiçoamentos anteriormente mencionados sem se afastar do escopo da presente invenção.

[00042] Será entendido que certas características e combinações secundárias são de utilidade e podem ser empregadas sem referência a outras características e combinações secundárias e são consideradas como estando dentro do escopo das reivindicações. Nem todas as etapas relacionadas nas várias figuras precisam ser realizadas na ordem específica descrita.

REIVINDICAÇÕES

1. Método para converter uma pluralidade de pequenas páginas associadas a um ou mais processos em execução em um sistema de computador em uma página grande, cada uma da pluralidade de páginas pequenas sendo associada a uma de uma pluralidade de entradas de tabela de páginas a partir de um sistema de tabela de página hierárquica contendo pelo menos dois níveis de tabelas de páginas, **caracterizado pelo fato de que** compreende as etapas de:

digitalizar (501) um último nível do sistema de tabela de páginas hierárquico para uma tabela de páginas na qual cada um de pelo menos um limiar de uma pluralidade de entradas está associado a uma pluralidade de páginas, resultando na identificação de uma tabela de páginas candidatas;

localizar (504) um segmento de memória composto por uma pluralidade de segmentos contíguos de memória física, suficientemente grande para armazenar cada um da pluralidade de segmentos de memória física associada a toda a pluralidade de entradas da tabela de páginas candidatas;

copiar (505) cada um da pluralidade de segmentos de memória física associados a toda a pluralidade de entradas da tabela de páginas candidatas no segmento de memória composto por uma pluralidade de segmentos contíguos de memória física; e

ajustar (506) uma entrada de tabela de página em uma tabela de página um nível antes do último nível do sistema de tabelas de página hierárquico a ser associado ao segmento de memória constituído por uma pluralidade de segmentos contíguos de memória física.

2. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** digitalizar um último nível do sistema de tabelas de páginas hierárquicas compreende digitalizar seletivamente cada um da pluralidade de espaços de endereços associados aos um ou mais pro-

cessos em execução no sistema de computador.

3. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** o limiar de uma pluralidade de entradas é toda a pluralidade de entradas.

4. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** cada uma da pluralidade de entradas da tabela de páginas candidatas está associada a um único dos um ou mais processos em execução no sistema de computador.

5. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** localizar um segmento de memória composto por uma pluralidade de segmentos contíguos de memória física compreende copiar dados a partir de um primeiro local perto de uma área de memória física para um segundo local longe da área para criar uma pluralidade de segmentos contíguos de memória física grande o suficiente para armazenar cada um da pluralidade de segmentos de memória física associada a toda a pluralidade de entradas da tabela de páginas candidatas.

6. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** o segmento de memória composto por uma pluralidade de segmentos contíguos de memória física está em um limite de byte predeterminado.

7. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** copiar cada um da pluralidade de segmentos da memória física associada a toda a pluralidade de entradas da tabela de páginas candidatas compreende ainda libertar a pluralidade de segmentos da memória física depois de terem sido copiados.

8. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** a entrada da tabela de páginas em uma tabela de páginas um nível antes do último nível do sistema de tabela de páginas hierárquicas foi anteriormente associado à tabela de páginas can-

didatas.

9. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** ajustar uma entrada de tabela de páginas compreende ainda libertar um segmento de memória contendo a tabela de páginas candidatas.

10. Sistema de computador **caracterizado pelo fato de que** compreende meios adaptados para executar as etapas de:

digitalizar (501) um último nível de um sistema de tabelas de páginas hierárquico contendo pelo menos dois níveis de tabelas de páginas, em cada um de uma pluralidade de espaços de endereços associados a um ou mais processos em execução em um sistema de computador para uma tabela de páginas no qual cada um de pelo menos um limiar de uma pluralidade de entradas está associado a um de uma pluralidade de segmentos de memória física, resultando na identificação de uma tabela de páginas candidatas;

localizar (504) um segmento de memória composto por uma pluralidade de segmentos contíguos de memória física, suficientemente grande para armazenar cada um da pluralidade de segmentos de memória física associada a toda a pluralidade de entradas da tabela de páginas candidatas;

copiar (505) cada um da pluralidade de segmentos de memória física associados a toda a pluralidade de entradas da tabela de páginas candidatas no segmento de memória composto por uma pluralidade de segmentos contíguos de memória física; e

ajustar (506) uma entrada de tabela de página em uma tabela de página um nível antes do último nível do sistema de tabelas de páginas hierárquicas, a entrada de tabela de página sendo associada previamente com a tabela de página candidata, a ser associada ao segmento de memória composto por uma pluralidade de segmentos contíguos da memória física.

11. Sistema, de acordo com a reivindicação 10, **caracterizado pelo fato de que** compreende meios adaptados para realizar uma etapa de liberar (505) um segmento de memória contendo a tabela de páginas candidatas.

12. Sistema, de acordo com a reivindicação 10, **caracterizado pelo fato de que** compreende meios adaptados para executar as etapas de:

receber (601) uma indicação de um subsistema de memória incapaz de trocar páginas grandes, indicando que um segmento de uma página grande deve ser trocado;

criar (602) uma nova tabela de páginas com cada entrada na nova tabela de páginas associada a um segmento da página grande; e

ajustar (604) uma entrada de tabela de página em uma tabela de página um nível antes do último nível do sistema de tabela de página hierárquica a ser associado com a nova tabela de página, em que a entrada de tabela de página em uma tabela de página um nível antes do último nível do sistema de tabelas de página hierárquico era anteriormente associado à página grande.

13. Sistema, de acordo com a reivindicação 12, **caracterizado pelo fato de que** o sistema de tabela de páginas hierárquico contém quatro níveis e/ou em que o sistema de tabela de páginas hierárquico da tabela de páginas é endereçado de acordo com uma arquitetura de 64 bits.

14. Sistema, de acordo com a reivindicação 12, **caracterizado pelo fato de que** cada página pequena é de 4KB, cada página grande é de 2MB, e/ou o segmento de memória composto por uma pluralidade de segmentos contíguos de memória física está em um limite de byte predeterminado de 2MB.

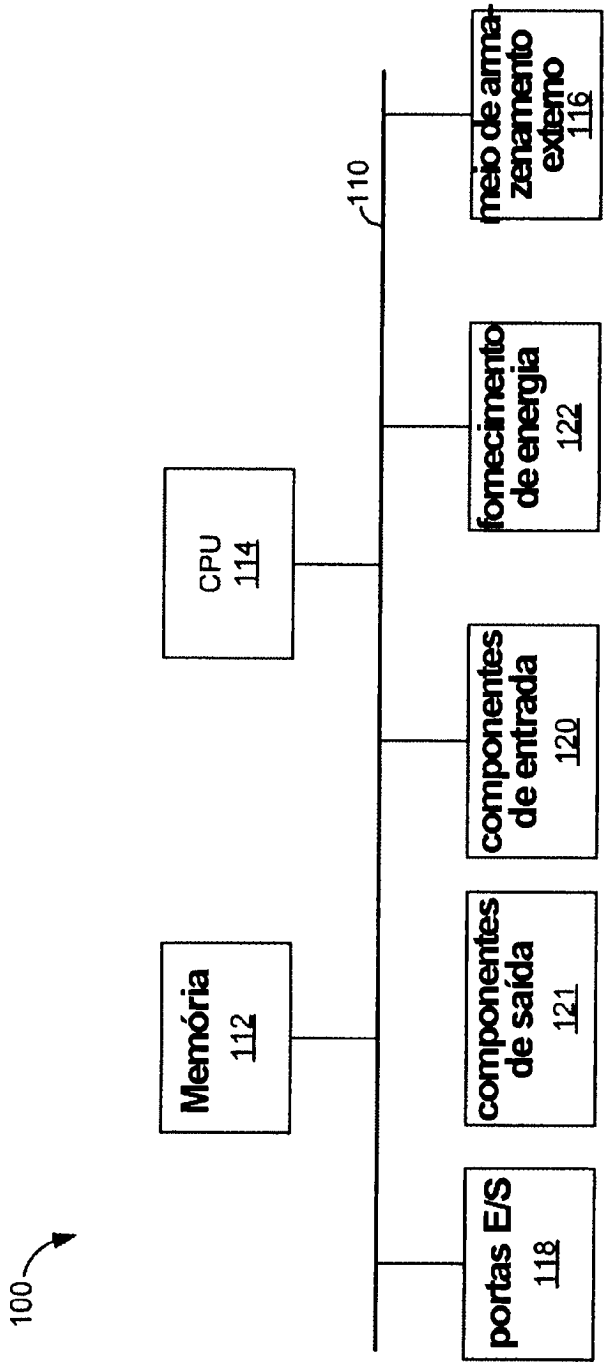
100
110
Memória
112
CPU
114
portas E/S
118
componentes de saída
121
componentes de entrada
120
fornecimento de energia
122
meio de armazenamento externo
116

FIG. 1.

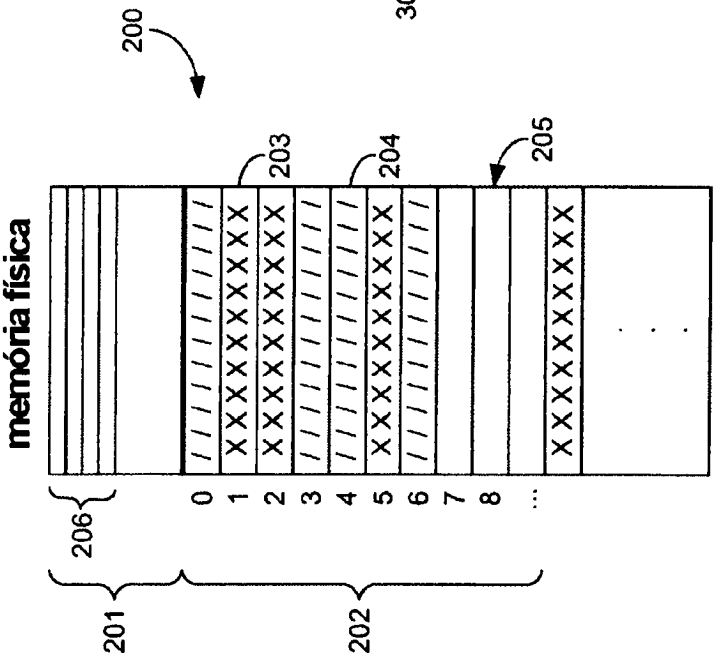
memória física
200
201
202
206
0
1
2
3
4
5
6
7
8
203
204
205

FIG. 2.

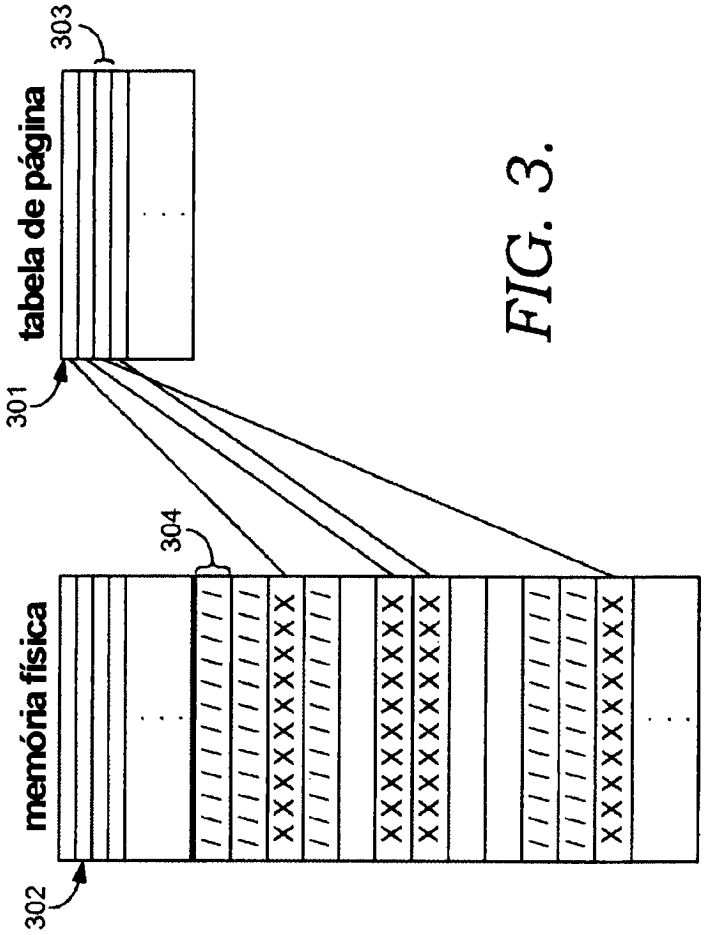
tabela de página
303
301
memória física
304
302

FIG. 3.

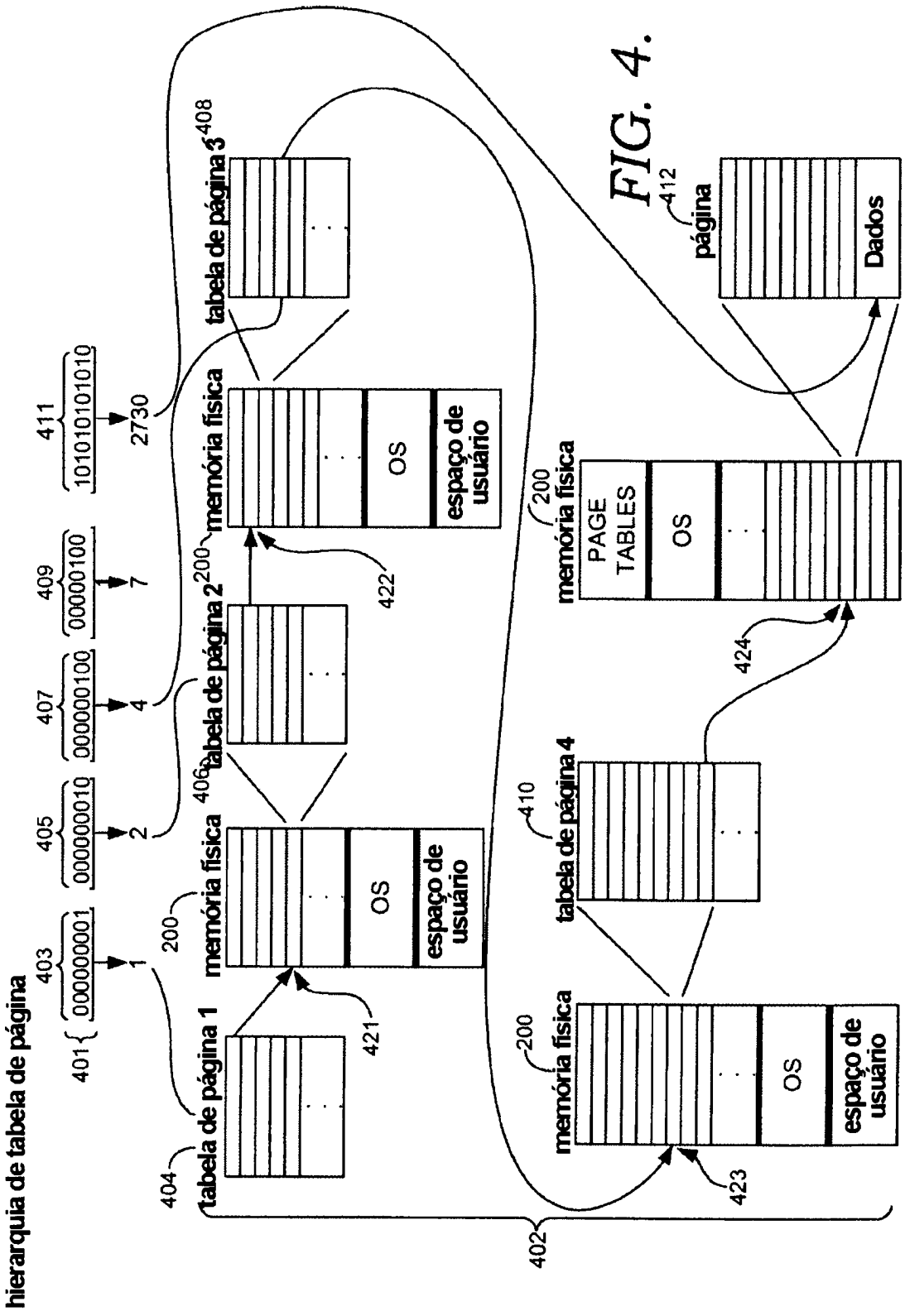
hierarquia de tabela de página
401
403
000000001
1
405
000000010
2
407
000000100
4
409
00000100
7
411
1010101010
2730
tabela de página 1
404
memória física
200
OS
espaço de usuário
421
tabela de página 2
406
422
tabela de página 3
408
402
tabela de página 4
410
424
PAGE TABLES
423
página
412
Dados

FIG. 4.

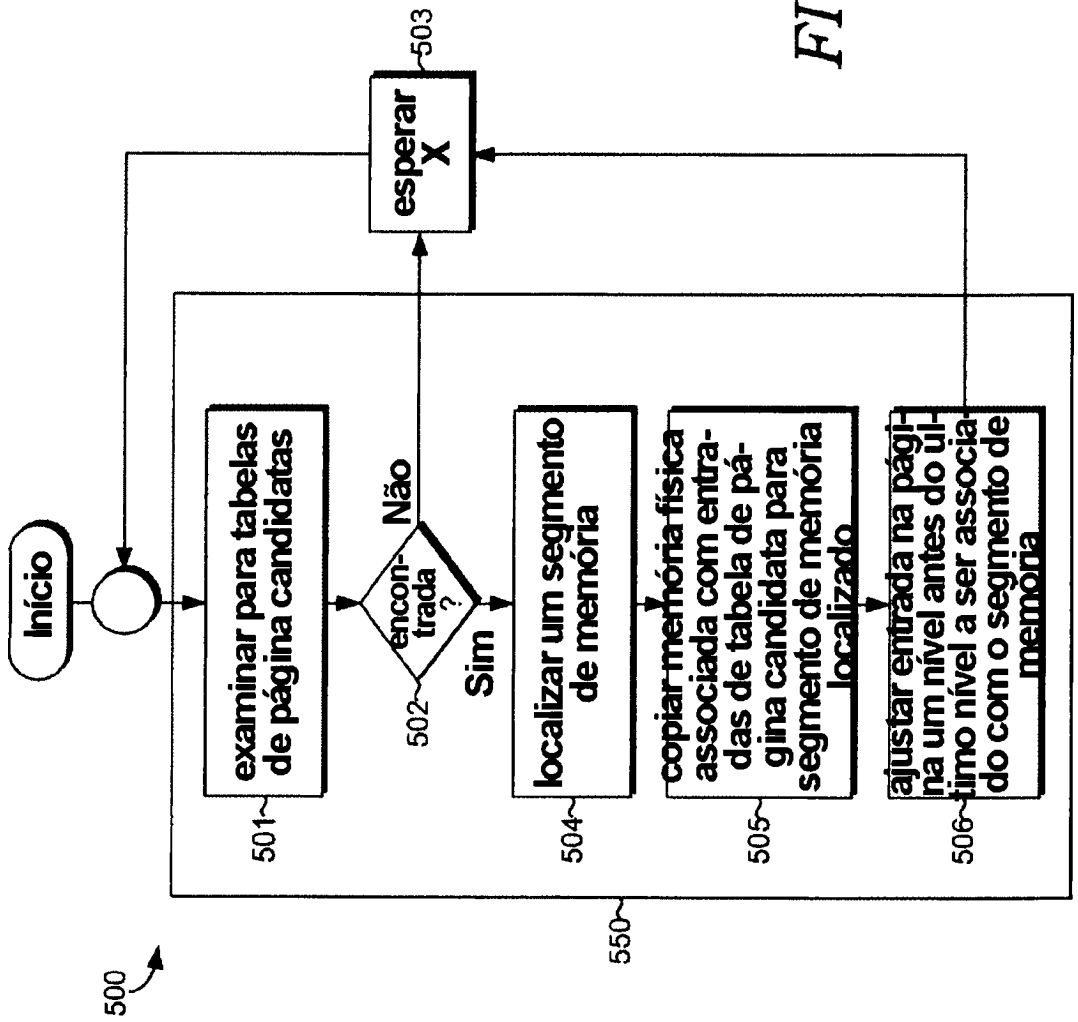
Início
examinar para tabelas de página candidatas
encontrada ?
Não
Sim
esperar X
localizar um segmento de memória
copiar memória física associada com entradas de tabela de página candidata para segmento de memória localizado
ajustar entrada na página um nível antes do último nível a ser associado com o segmento de memória
501
502
503
504
505
506
550
500

FIG. 5.

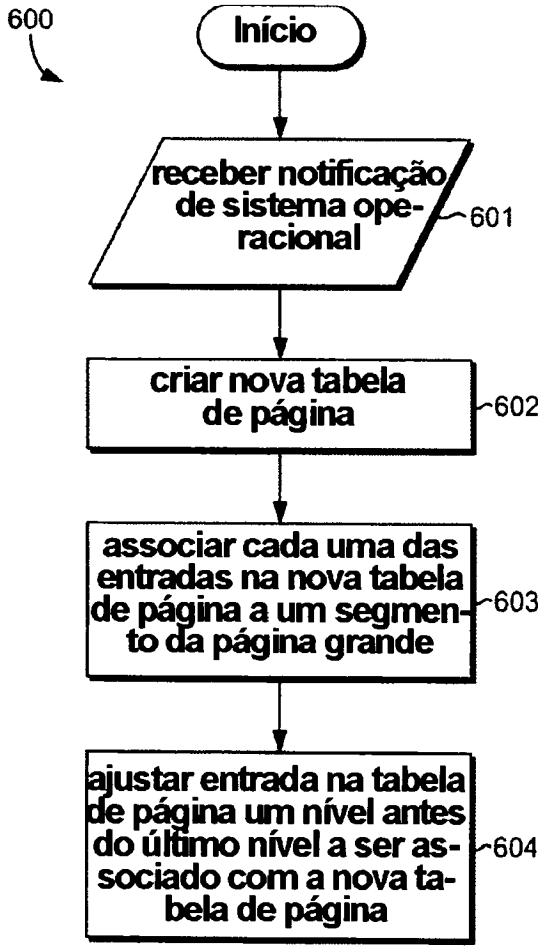
600
Início
receber notificação de sistema operacional
601
criar nova tabela de página
602
associar cada uma das entradas na nova tabela de página a um segmento da página grande
603
ajustar entrada na tabela de página um nível antes do último nível a ser associado com a nova tabela de página
604

FIG. 6.

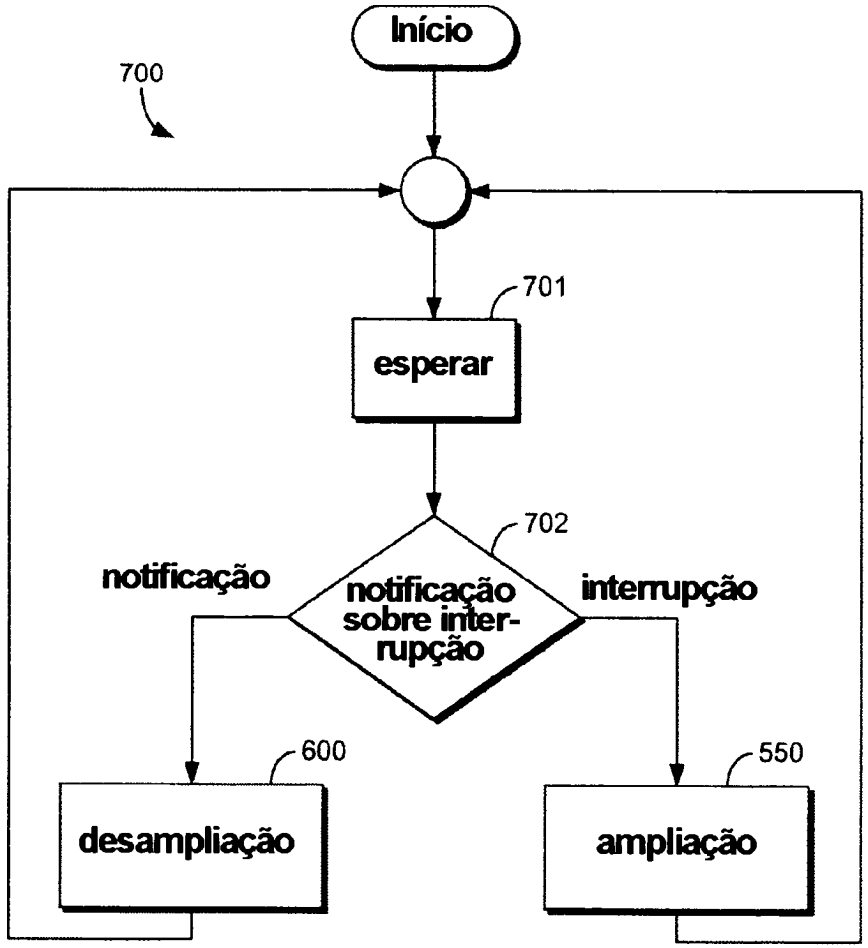
700
Início
701
esperar
702
notificação sobre interrupção
notificação
interrupção
600
desampliação
550
ampliação

FIG. 7.