**1 Processador**

É um componente especializado que tem a função de acelerar, endereçar, resolver ou preparar dados, dependendo da aplicação. Basicamente é uma poderosa máquina de calcular: Ela receber um determinado volume de dados,orientados em padrão binário 0 e 1 e tem a função de responder a esse volume,processando a informação com base em instruções armazenadas em sua memória interna.

**1.2 O que é ULA?**

ULA é a sigla para Unidade Lógica Aritmética trata-se do circuito que se encarrega de realizar as operações matemáticas requisitadas por um determinado programa. Ela soma,subtrair,divide,determina se um número é positivo ou negativo ou se é zero.Além de executar funções aritméticas , uma ULA deve ser capaz de determinar se uma quantidade é menor ou maior que outra .A ULA pode executar funções lógicas com letras e com números.

**1.3 Unidade de Controle**

É responsável por gerar todos os sinais que controlam as operações no exterior do CPU, e ainda por dar todas as instruções para o correto funcionamento interno do processador. A unidade de controle executa três ações básicas intrínsecas e pré-programadas pelo próprio fabricante do processador,são elas:Busca(fetch),decodificação e execução.Assim sendo,todo processador,ao iniciar sua operação , realiza uma operação cíclica , tendo como base essas três ações . Dependendo do tipo de microprocessador, a unidade de controle pode ser fixa ou programável. A unidade fixa é aquela unidade que já vem com todo o conjunto de instrução programado em uma PLA que é construída pelo fabricante,dentro da UC.

**1.4 Cache**

É um dispositivo de acesso rápido, interno a um sistema, que serve de intermediário entre um operador de um processo e o dispositivo de armazenamento. Esses dados estão rapidamente acessíveis e não é necessário executar uma varredura em disco ou na RAM para buscar as informações.A cachê se baseia fundamentalmente nos princípios de localidade temporal e espacial.Funcionando como um elemento intermediário entre a CPU e a Memória Principal e armazenando as informações que muito provavelmente serão requisitadas pela CPU.

**1.4.1 Níveis de Memória Cache**

Com o aumento crescente da velocidade da CPU e visando minimizar um grande impacto no custo da Memória Cache os fabricantes vêm estabelecendo diferentes níveis de memória cachê.

* Cache L1 uma pequena porção de memória estática presente dentro do processador. Em alguns tipos de processador é dividido em dois níveis:dados e instruções.
* Cache L2 Possuindo o Cache L1 um tamanho reduzido e não apresentando uma solução ideal, foi desenvolvido o cachê L2, que contém muito mais memória que o cachê L1. Ela é mais um caminho para que a informação requisitada não tenha que ser procurada na lenta memória principal.
* Cache L3 Terceiro nível de cachê de memória. Inicialmente utilizado pelo AMD K6-III utilizava o cachê externo presente na placa-mãe como uma memória de cachê adicional.Ainda é um tipo de cachê raro devido a complexidade dos processadores atuais,com suas áreas chegando a milhões de transistores por micrometros ou nanômetros de área.

**1.5 Registradores**

É uma unidade de memória capaz de armazenar n bits. Os registradores estão no topo da hierarquia de memória,sendo assim,são o meio mais rápido e caro de se armazenar um dado.Lembrando que os registradores são circuitos digitais capazes de armazenar e deslocar informações binárias, e são tipicamente usados como dispositivo de armazenamento temporário.São utilizados na execução de programas de computadores ,disponibilizando um local para armazenar dados.Na maioria dos computadores modernos,quando da execução das instruções de um programa,os dados são movidos da memória principal para os registradores.Então , as instruções que utilizam estes dados são executadas pelo processador e ,finalmente,os dados são movidos de volta para a memória principal.

**1.6 Memory Management Unit(MMU)**

O Memory Management Unit (MMU) é o responsável pela coordenação do funcionamento da memória. O processador só pode ser rápido se a memória RAM acompanhar.O MMU é o recurso que transforma as instruções lógicas(virtuais) em endereços físicos nos bancos de memória.O processador varre a memória atrás de dados e instruções e o MMU é o recurso que anota aonde cada informação do sistema está hospedada na memória.É ele que diz onde o processador deve procurar.

**1.7 Clock**

O clock é responsável por medir os ciclos e orienta o ritmo do fluxo de troca de informações no processador , é um dos principais critérios para estabelecer a velocidade do processador.Vale Ressaltar,no entanto , que outros pontos entram nesta conta,como interface de memória , quantidade de cachê,arquitetura , entre outros.

**1.8 Soquetes**

Os soquetes são entradas que as placas-mãe possuem para instalação dos processadores.Existem diversos tipos,cada um deles é específicos para cada fabricante ou modelo da unidade de processamento que será instalada.

**2.1 Tipos de Processadores**

Como os soquetes,os processadores apresentam diversos tipos , que variam de acordo com a quantidade de núcleos , capacidade de processamento e suporte ao overclock,prática de aumentar a capacidade do chip.

* **Single-core ou multi-core**: Esta características indica a quantidade de núcleos de processamento que um processador pode ter,podendo variar de apenas um núcleo até mais de oito núcleos.Quando maior o número de cores,maior é a capacidade de processar tarefas simultaneamente e acelerar as aplicações do PC.
* **Arquitetura 32 ou 64 bits**:Essa característica remete á capacidade de processamento de informações do processador.Apenas chips com arquitetura de 64 bits é possível que o PC aproveite quantidades superiores a 3GB de RAM,além de processarem blocos maiores de dados de maneira mais rápida.
* **Compatibilidade com overclock** : Esta é uma técnica utilizada para aumentar a velocidade nominal do processador e conceder um desempenho extra ao usuário.Processadores que possuem esta capacidade são identificados como “Unlocked”, no caso de Intel , ou “Black Edition” , para AMD.

**3 Introdução a Arquitetura RISC**

A arquitetura RISC é constituída por um pequeno conjunto de instruções simples que são executadas diretamente pelo hardware,sem a intervenção de um interpretador(microcódigo),ou seja , as instruções são executadas em apenas uma microinstrução.Risc é também a arquitetura adotada para os processadores dos videogames modernos,que proporcionam um hardware extremamente dedicado somente á execução do jogo , tornando- o muito mais rápido em relação à micro computadores com mais recursos,embora com processador x86.

**3.1 Características da Arquitetura RISC**

Existe um conjunto de características que permite uma definição de arquitetura básica RISC.

* **Menor quantidade de instruções** : Talvez a características mais marcante das arquiteturas RISC,seja a de possuir um conjunto de instruções menor(todas também com largura fixa).Vem daí o nome dado a arquitetura RISC(computadores com um conjunto reduzido de instruções).Com o conjunto de instruções reduzido e cada uma delas tendo suas funções otimizadas,os sistemas possuíam um resultado melhor em questão de desempenho.Em virtude do conjunto reduzido das instruções , acarretavam em programas um pouco mais longos.
* **Execução otimizada de chamadas de função** : Outra evolução da arquitetura RISC tem relação com a chamada de retinas e passagem de parâmetros .Estudos indicam que as chamadas de funções consomem um tempo significativo de processador.Elas requerem poucos dados,mas demoram muito tempo nos acessos a memória.
* **Modo de Execução com Pipelining**: Uma das características mais relevantes da arquitetura RISC é o uso de pipelining,mesmo sabendo que ela tem um funcionamento mais efetivo quando as instruções são todas bastante parecidas.Imaginando estágios de uma linha de montagem,não e interessantes que um estágio termine antes do outro , pois nesse caso perde-se a vantagem da linha de montagem.O objetivo de cada instrução , é completar um estágio de pipeline em um ciclo de clock,mas esse objetivo nem sempre é alcançado.

**4 Arquitetura CISC**

CISC (sigla para Complex Instruction Set Computer , ou , “Computador com um Conjunto Complexo de Instruções”) é uma linha de arquitetura de processadores capaz de executar centenas de instruções complexas diferentes sendo,assim,extremamente versátil.Exemplos de processadores CISC são os 386 e os 486 da Intel.Os processadores baseados na computação de conjuntos de instruções complexas contêm um micro-programação,ou seja,um conjunto de códigos de instruções que são gravados no processador, permitindo-lhe receber as instruções dos programas e executá-las,utilizando as instruções contidas na sua micro-programação.Seria como quebrar esta instruções , já em baixo nível, em diversas instruções mais próximas do hardware(as instruções contidas no microcódigo do processador).Como características marcante esta arquitetura contém um conjunto grande de instruções , a maioria deles em um elevado grau de complexidade.

**4.1 Características da Arquitetura CISC**

Existe um conjunto de características que permite uma definição de arquitetura básica CISC: são elas:

**Diferenças de velocidade entre memória e processador** – no final da [década de 1970](https://pt.wikipedia.org/wiki/D%C3%A9cada_de_1970), a IBM verificou que essa diferença era um problema em seus sistemas, algumas operações eram realizadas por programas, acarretando muitos acessos a uma memória lenta. A solução encontrada foi criar novas instruções de máquina para executar tais operações, podendo-se acreditar que esse foi o início do aumento da quantidade de instruções no CISC.

**Emprego de microcódigo** – o surgimento e a real vantagem de custo/beneficio do emprego de microcódigo sobre programação diretamente no hardware induziram os projetistas a criar mais e mais instruções, devido a facilidade e a flexibilidade decorrentes. Desenvolvimento acelerado de linguagens de alto nível – na [década de 1980](https://pt.wikipedia.org/wiki/D%C3%A9cada_de_1980), havia um crescimento acelerado do emprego de linguagens de alto nível, o que conduzia os projetistas de processadores a incluir cada vez mais instruções de máquinas em seus produtos, como o propósito de manter um suporte adequado na compilação.

**Densidade de código a ser executado** – as arquiteturas CISC procuram obter um código compacto após a compilação, de modo a não consumir memória em excesso. Isso era necessário em uma época em que as memórias eram caras e de reduzindo tamanho. Construindo conjuntos de instruções, cada uma delas mais próxima do significado do comando de alto nível, poder-se-ia obter códigos executáveis mais densos, mais compactos. Alega Patterson que isto acarretaria também mais bits nas instruções (códigos de operações com mais bits devido à quantidade delas, bem como mais modos de endereçamento), o que contrabalançaria aquela pretensa vantagem.

**Necessidade de compatibilidade com processadores anteriores** – uma das metas sempre seguida pela Intel e outros fabricantes foi a de conservar a compatibilidade entre as versões de seus processadores. Assim o processador [486](https://pt.wikipedia.org/wiki/486) veio com apenas algumas instruções novas e todo o código do [386](https://pt.wikipedia.org/wiki/386) junto, códigos executáveis para o 386 rodavam também no 486, e os usuários poderiam trocar de computador sem nenhum custo adicional de compilação, etc. O mesmo aconteceu com o [Pentium I](https://pt.wikipedia.org/wiki/Pentium_II), II, III e 4. Mesmo isso, embora seja um notório requisito importante de marketing, acarreta uma limitação especificação de novas arquiteturas. Dessa forma, as arquiteturas novas só crescem em quantidade de instruções, visto que o fabricante nunca retira as instruções antigas devido ao problema de compatibilidade.

**5 Diferença entre Arquitetura RISC x Arquitetura CISC**

As diferenças podem ser notadas pelas características de cada arquitetura:

|  |  |
| --- | --- |
| ARQUITETURA RISC | ARQUITETURA CISC |
| Controle por Hardware | Controle Microprogramado |
| Pequeno conjunto de instruções | Instruções de dois operando ADD CX |
| Todas as instruções tem tamanho fixo | Modos registro-registro,registro-memória |
| Execução otimizada de chamada de funções | Múltiplos modos de endereçamento á memória,incluindo indexação |
| Pouquíssimos modos de endereçamento | Instruções de largura(tamanho) variável,conforme modo de endereçamento utilizado |
| Uso intenso de pipeline | Instruções requerem múltiplos ciclos de máquina para execução,variando também com o modo de endereçamento. |
| Execução rápida de cada instrução | Poucos registros |

**6 Arquitetura x86**

A arquitetura x86 é o nome genérico dada á família de processadores baseados no Intel 8086, da Intel Corporation. A arquitetura é chamada x86 porque os primeiros processadores desta família eram identificados somente por números terminados com a sequência "86": o [8086](https://pt.wikipedia.org/wiki/Intel_8086), o [80186](https://pt.wikipedia.org/wiki/Intel_80186), o [80286](https://pt.wikipedia.org/wiki/Intel_80286), o [80386](https://pt.wikipedia.org/wiki/Intel_80386) e o [80486](https://pt.wikipedia.org/wiki/Intel_80486). Como não se pode ter uma marca registrada sobre números, a Intel e a maior parte de seus competidores começaram a usar nomes que pudessem ser registrados como marca, como [Pentium](https://pt.wikipedia.org/wiki/Pentium) para as gerações de [processadores](https://pt.wikipedia.org/wiki/Processadores) posteriores, mas a antiga nomenclatura tinha forjado um termo para toda a [família](https://pt.wikipedia.org/wiki/Fam%C3%ADlia). O termo x86 não é sinônimo de compatibilidade com IBM PC, pois isso sugere uma infinidade de outros hardwares e sistemas embarcados bem como [computadores](https://pt.wikipedia.org/wiki/Computadores) usando chips x86 antes do moderno de mercado ser compatível com PC, alguns deles antes mesmo do IBM PC em si. Houve várias tentativas, dentro da [Intel](https://pt.wikipedia.org/wiki/Intel) em si, para quebrar o domínio do mercado da arquitetura x86, que descendia diretamente do simples [microprocessadores](https://pt.wikipedia.org/wiki/Microprocessadores) de 8 bits. Exemplos disso são as IAPX 432 (aliás, Intel 8800), o Intel 960, Intel 860 e Intel e a arquitetura do Hewlett Packard Itanium. No entanto, o aperfeiçoamento contínuo de 86 micro arquiteturas, circuitos, e fabricação de [semicondutores](https://pt.wikipedia.org/wiki/Semicondutores) provaria que é difícil substituir x86 em vários segmentos. O 64, da extensão bit da AMD x86 (que a Intel finalmente respondeu, com um[design](https://pt.wikipedia.org/wiki/Design) compatível) e a escala de processadores x86, como o eight-core Intel Xeon e 12-core AMD Opteron destaca-se o x86 como um exemplo de contínuo aperfeiçoamento e como as normas estabelecidas pela [indústria](https://pt.wikipedia.org/wiki/Ind%C3%BAstria) pode resistir à concorrência das novas arquiteturas.

**6.1 Propriedades Básicas Da Arquitetura x86**

A arquitetura x86 é um comprimento variável de instrução, primeiramente são modelados dois endereços “CISC” com [ênfase](https://pt.wikipedia.org/wiki/%C3%8Anfase) em compatibilidade com versões anteriores. No entanto o conjunto de instruções não é típico CISC, mas, basicamente, uma versão estendida e [ortogonal](https://pt.wikipedia.org/wiki/Ortogonal) de simples oito bits, das arquiteturas 8008, 8080 e 8085. O [Byte](https://pt.wikipedia.org/wiki/Byte) de endereçamento é suportado e as palavras são armazenadas na memória com ordem de little-endian byte. O acesso à memória para endereços desalinhados é permitido em todos os tamanhos de palavras suportadas. O maior tamanho nativo para [aritmética](https://pt.wikipedia.org/wiki/Aritm%C3%A9tica) de inteiros e endereços de memória (ou deslocamentos) é de 16, 32 ou 64 bits dependendo da geração da [arquitetura](https://pt.wikipedia.org/wiki/Arquitetura) (processadores mais recentes incluem um suporte direto para inteiros menores também). Vários valores escalares podem ser tratados simultaneamente, através da unidade SIMD presente em gerações posteriores, conforme descrito abaixo. Compensações imediatas de endereçamento e dados de imediato podem ser expressos em quantidades de 8 bits conforme a frequência que ocorrem os casos ou contextos em que o [intervalo](https://pt.wikipedia.org/wiki/Intervalo) de 128. 127 é o suficiente. Instruções típicas são, portanto, de 2 ou 3 [bytes](https://pt.wikipedia.org/wiki/Bytes) de comprimento (embora alguns são muito mais longos, e algumas são de [byte](https://pt.wikipedia.org/wiki/Byte) único). Para melhor conservar o espaço de codificação, a maioria dos registros são expressos em opcodes usando três bits, e no máximo um operando em uma instrução que pode ser um local de memória (alguns projetos "CISC" altamente [ortogonais](https://pt.wikipedia.org/wiki/Ortogonal), como o PDP-11, podem utilizar dois). No entanto, esta [memória](https://pt.wikipedia.org/wiki/Mem%C3%B3ria) operando também pode ser de destinação (ou a combinação entre fonte e destinação), enquanto o outro operando, a fonte, pode ser tanto registrada quanto imediata. O [número](https://pt.wikipedia.org/wiki/N%C3%BAmero) relativamente pequeno de registros gerais (também herdados de seus antecessores de 8 bits) fez-se endereçamento de registro relativo (usando pequenas compensações imediatas) como um importante método de acesso a operando, especialmente na [pilha](https://pt.wikipedia.org/wiki/Pilha). Muito trabalho foi, portanto, investido na [fabricação](https://pt.wikipedia.org/wiki/Fabrica%C3%A7%C3%A3o) desses acessos de modo tão rápido como registrar os acessos, ou seja, um ciclo de transferência de instrução, na maioria dos casos em que os dados acessados estão disponíveis no [cache](https://pt.wikipedia.org/wiki/Cache) de nível superior.

**6.2 As implementações Atuais**

Durante a execução, processadores x86 atuais empregam uma [decodificação](https://pt.wikipedia.org/wiki/Decodifica%C3%A7%C3%A3o) extra para dividir a maioria das instruções em pedaços menores (micro-operações). Estes são então entregues a uma unidade de controle que os tampões e os [cronogramas](https://pt.wikipedia.org/wiki/Cronograma) estão em conformidade de acordo com a x86-semântica, para que possam ser executadas, parcialmente em paralelo, por uma das várias unidades (mais ou menos especializadas) de execução. Quando foi introduzida, esta abordagem foi por vezes referida como um "núcleo RISC" ou como "tradução RISC", em parte por razões de [marketing](https://pt.wikipedia.org/wiki/Marketing), mas também porque essas micro-operações compartilham algumas propriedades com certos tipos de instruções RISC. No entanto, o [microcódigo](https://pt.wikipedia.org/w/index.php?title=Microc%C3%B3digo&action=edit&redlink=1) tradicional (usado desde a [década](https://pt.wikipedia.org/wiki/D%C3%A9cada) de 1950) também inerentemente compartilha muitas das mesmas propriedades; a nova abordagem difere, principalmente na medida em que a tradução para micro-operações agora ocorrem de forma [assíncrona](https://pt.wikipedia.org/wiki/Ass%C3%ADncrona). Não ter que sincronizar as unidades de execução com as etapas de decodificação abre possibilidades para uma análise mais aprofundada do fluxo de código (buffer) e, portanto, permite a detecção de operações que podem ser executadas em paralelo, alimentando simultaneamente mais de uma unidade de execução. Os mais recentes [processadores](https://pt.wikipedia.org/wiki/Processadores) também fazem o oposto quando for o caso, pois eles combinam certas sequencias x86 (como uma comparação seguida de um salto condicional) um processo mais complexo de micro-op, que se encaixa no modelo de execução melhor e, portanto, pode ser executado mais rápido ou com menos recursos da máquina envolvida. Outra forma de tentar melhorar o desempenho é o [cache](https://pt.wikipedia.org/wiki/Cache) decodificado de micro-operações, de modo que o [processador](https://pt.wikipedia.org/wiki/Processador) pode acessar diretamente o [decodificado](https://pt.wikipedia.org/w/index.php?title=Decodificado&action=edit&redlink=1) de micro-operações a partir de um cache especial, em vez de descodificá-lo novamente. A “Execution Trace Cache” é encontrada na microarquitetura da [Intel](https://pt.wikipedia.org/wiki/Intel) NetBurst (Pentium 4) é até agora o único exemplo generalizado desta técnica. Transmeta usa um método completamente diferente em seus [CPUs](https://pt.wikipedia.org/wiki/CPU) x86 compatíveis. Eles usam tradução para converter as instruções x86 para instruções nativas do [CPU](https://pt.wikipedia.org/wiki/CPU). Transmeta argumenta que sua abordagem permite projetos de energia mais eficientes, uma vez que a CPU pode abandonar a etapa de [decodificação](https://pt.wikipedia.org/wiki/Decodifica%C3%A7%C3%A3o) complicada para implementações x86 tradicionais.

**6.3 Registradores X86**

**Registro de 16 bits** O original Intel 8086 e 8088 tem quatorze registros de 16 bits. Quatro deles (AX, BX, CX, DX) são registradores de propósito geral (GPRS, embora cada um possa ter uma finalidade adicional: por exemplo, apenas CX pode ser usado como um [contador](https://pt.wikipedia.org/wiki/Contador) com a instrução de [loop](https://pt.wikipedia.org/wiki/Loop)). Cada um pode ser acessado como dois [bytes](https://pt.wikipedia.org/wiki/Bytes) separados (daí o byte mais alto BX pode ser acessado como BH e byte baixo como BL). Há dois registros de ponteiro: SP, que aponta para o topo da [pilha](https://pt.wikipedia.org/wiki/Pilha) e da BP (base pointer) que é usado para apontar para algum outro lugar na [pilha](https://pt.wikipedia.org/wiki/Pilha), geralmente acima de variáveis locais. Dois registros (SI e DI) são para a indexação de matriz.

Quatro registradores de segmento (CS, DS, SS e ES) são usados para formar um endereço de memória. O registrador de flags contém sinalizadores como de proceder com tais flags, overflow de flag e underflow de flag. Finalmente, o ponteiro de instrução (IP) aponta para a próxima instrução que será obtida a partir da memória e depois executados.

Na Intel 80286, três registros especiais contém endereços da tabela descrita (GDTR, LDTR, IDTR), e um registro de quarta tarefa (TR).

**Registro 32 bits** Com o advento do [processador](https://pt.wikipedia.org/wiki/Processador) de 32 bits 80386, os registradores de 16 bits, registradores base, registradores de índice, ponteiro de instrução, e registradores de flag, mas não os registradores de segmento, foram ampliados para 32 bits. Este é representado por um [prefixo](https://pt.wikipedia.org/wiki/Prefixo) "E" (para Extended) para registrar os nomes em [linguagem assembly](https://pt.wikipedia.org/wiki/Assembly) x86. Assim, o registro AX diferencia o registrador de 16 bits do novo EAX 32-bit, SI corresponde ao menor de 16 bits do ESI, e assim por diante. Os [registradores](https://pt.wikipedia.org/wiki/Registradores) de uso geral, registradores de base e registradores de índice podem ser usados como base em modos de endereçamento, e todos os registros, exceto o ponteiro de [pilha](https://pt.wikipedia.org/wiki/Pilha) pode ser utilizado como [índice](https://pt.wikipedia.org/wiki/%C3%8Dndice) de endereçamento.

Dois novos registros de segmento (FS e GS) foram adicionados. Com um maior número de registros, instruções e operandos, o formato do binario|código de [máquina](https://pt.wikipedia.org/wiki/M%C3%A1quina) foi alargado. Para fornecer compatibilidade com versões anteriores, os [segmentos](https://pt.wikipedia.org/wiki/Segmentos) com código executável pode ser marcado como contendo ou 16 bits ou 32 bits de instruções. Prefixos especiais permitem a inclusão de instruções de 32 bits em um segmento de 16 bits, ou vice-versa.

Com o 80486 uma unidade de processamento de ponto flutuante (FPU) foi adicionada, com oito registros de 80 bits de largura.

Com o Pentium II, oito [registradores](https://pt.wikipedia.org/wiki/Registradores) de 64 bits MMX foram adicionados. Com o Pentium III, um de 32 bits Streaming SIMD Extension (SSE) de controle / registrador de status (mxcsr) e oito de 128 bits SSE [registradores](https://pt.wikipedia.org/wiki/Registradores) de ponto flutuante (XMM0. XMM7) foram adicionados [17]. Com o Pentium III, um Streaming SIMD Extension (SSE) de 32 bits, um registrador de controle (MXCSR) e oito registradores SSE de pontos flutuantes de 128 bits (XMM0. XMM7) foram adicionados.

**Registro 64 bits** Com o [processador](https://pt.wikipedia.org/wiki/Processador) AMD Opteron, a [arquitetura](https://pt.wikipedia.org/wiki/Arquitetura) x86 estendeu os registradores de 32 bits para registradores de 64 bits de uma forma similar como a extensão de 16 bits para 32 bits foi feita, foram adicionadas oito registradores de 64 bits (R8, R9 R15.), entre outras. No entanto, essas extensões são apenas utilizáveis em modo de 64 bits. Os modos de endereçamento não mudaram drasticamente a partir do modo de 32 bits, com exceção que a [resolução](https://pt.wikipedia.org/wiki/Resolu%C3%A7%C3%A3o) foi estendida para 64 bits, endereços [virtuais](https://pt.wikipedia.org/wiki/Virtual) foram estendidos para 64 bits, a fim de impedir endereços [virtuais](https://pt.wikipedia.org/wiki/Virtual) que se têm revelado problemáticos em outras [arquiteturas](https://pt.wikipedia.org/wiki/Arquitetura), e mais alguns detalhes foram drasticamente reduzidos.