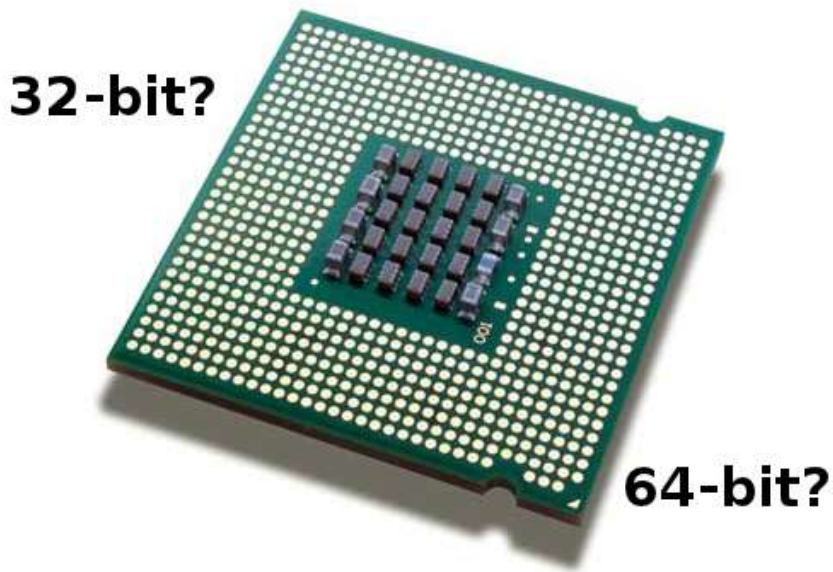
32-бітові та 64-бітові процесори
Порівняльні характеристики



Найпоширенішими і найуспішнішими донедавна були процесори з архітектурою IA32, яка була введена з появою покоління процесорів i80386 на заміну 16-бітним 8086, 80186, 80286.

Досить вдале 64-бітове розширення класичної 32-бітової архітектури IA32 було запропоноване в 2002 році компанією AMD (спочатку називалося x86-64, зараз - AMD64) в процесорах сімейства K8. Через деякий час компанією Intel було запропоновано власне позначення - EM64T (англ. *Extended Memory 64-bit Technology*). Суть нової архітектури одна і та ж: розрядність основних внутрішніх регістрів 64-бітових процесорів подвоїлася (з 32 до 64 біт), а 32-бітові команди x86-кода отримали 64-бітові аналоги. Крім того, за рахунок розширення розрядності шини адрес обсяг пам'яті, що адресується процесором, істотно збільшився.

Opteron , перший процессор, у якому введено x86-64 розширення в 2003р.

x86-64 (також відомий **як x64, x86_64 та amd64**) є 64-бітний варіант набору інструкцій x86. Він підтримує значно більшу кількість (теоретично, 2^{64} байт або 16 exbibytes) віртуальної пам'яті і фізичної пам'яті, порівняно з 32-розрядними попередниками. x86-64 має також 64-бітові регістри загального призначення і інші численні удосконалення. Вихідна специфікація була створена AMD, і була реалізована AMD, Intel , VIA , та ін.. Він повністю сумісний з 16-бітним і 32-бітним кодом x86.

Специфікація X86-64 відрізняється від Intel Itanium (раніше IA-64), архітектури, яка не сумісна з рідним набором команд з архітектурою x86.

Набір інструкцій AMD64 реалізований в AMD, Opteron, Athlon 64 , Athlon 64 FX , Athlon 64 X2 , Athlon II , Athlon X2, Turion 64 , Turion 64 X2 , через Sempron , Phenom , Phenom II , FX , і Fusion процесорах.

Архітектурні особливості

Основною визначальною характеристикою AMD64 є наявність 64-бітових регістрів загального призначення (наприклад, RAX і RBX), арифметичних і логічних операцій над 64-бітними числами, а також 64-бітових віртуальних адрес.

64-розрядні цілі числа

Всі регістри загального призначення (GPRS) розширені з 32 біт до 64 біт, і всі арифметичні і логічні операції, пам'ять-регістр і регістр-пам'ять і т.д., тепер можуть працювати безпосередньо з 64-розрядними цілими числами. Заштовхування і виштовхування у стек можливе 8 байтовими кроками, і вказівники мають розмір 8 байт.

Додано регістри

На додаток до збільшення розміру регістрів загального призначення, кількість регістрів загального призначення **збільшується** з восьми (тобто EAX, EBX, ECX, EDX, EBP, ESP, ESI, EDI) в x86 **до 16** (тобто RAX , RBX, RCX, RDX, RBP, RSP, RSI, RDI, R8, R9, R10, R11, R12, R13, R14, R15). Таким

чином, можна зберегти більше локальних змінних в регістрах, а не в стеку, і утримувати в регістрах часто використовувані константи; Аргументи для невеликих і швидких підпрограм також можуть бути передані в регістрах більшою мірою.

AMD64 має менше регістрів, ніж багато RISCподібних (які зазвичай мають 32 регістри) або VLIW машин - таких як IA-64 (який має 128 регістрів). Проте, реалізація AMD64 може мати набагато більше внутрішніх регістрів, ніж кількість архітектурних регістрів, доступних в наборі команд.

Додано XMM (SSE) регістри

Крім того, число 128-бітних регістрів XMM (використовується для Streaming SIMD інструкцій) також збільшилося з 8 до 16 (XMM0 — XMM15).

Великий віртуальний адресний простір

Архітектура AMD64 визначає формат віртуального адресного простору 64-бітовим, з яких молодші 48 біт використовуються в поточних реалізаціях. Це дозволяє мати **до 256 ТБ** (2^{48} байт) віртуального адресного простору. Визначена архітектура дозволяє використати цю межу в майбутніх реалізаціях повною мірою (до 64 біт), тобто розширення віртуального адресного простору для 16EB (2^{64} байт). Порівняйте з 4 ГБ (2^{32} байт) для x86.

Великий фізичний адресний простір

Оригінальна архітектура AMD64 реалізована на 40-розрядних фізичних адресах і тому може адресувати до 1 ТБ (2^{40} байт) оперативної пам'яті. Поточні реалізації архітектури AMD64 (починаючи з мікроархітектури AMD 10h) розширені до 48-бітних фізичних адрес і, отже, можуть адресувати до 256 Тбайт ОЗУ. Архітектурні можливості, що зараз розглядаються, це 52 біт в майбутньому (обмежується форматом запису таблиці сторінок); Це дозволить отримати до 4 PB оперативної пам'яті. Для порівняння, 32-розрядні процесори x86 обмежені 64 ГБ оперативної пам'яті в режимі продовженої фізичної адреси (PAE), або 4 Гб оперативної пам'яті без режиму PAE.

Великий фізичний адресний простір в стандартному режимі

При роботі в стандартному режимі архітектура AMD64 підтримує Physical Address Extension режим (PAE), як і більшість сучасних процесорів x86, але AMD64 поширюється PAE з 36 бітів архітектурної межі до 52 біт фізичної адреси.

Вказівник команд

Тепер Інструкції можуть посилатися на дані, які визначаються 64-бітовим вказівником команд (RIP регістр). Це робить Position Independent код, який часто використовується в більш ефективних поділюваних бібліотеках і кодах, завантажуваних під час виконання.

Інструкції SSE

SSE (англ. *Streaming SIMD Extensions*, потокове SIMD-розширення процесора) — це SIMD набір інструкцій, розроблених Intel, і вперше представлених у процесорах серії Pentium III як відповідь на аналогічний набір інструкцій 3DNow! відAMD, який був представлений роком раніше.

Оригінальна архітектура AMD64 прийняла від Intel SSE і SSE2 в якості основних інструкцій. Ці набори інструкцій забезпечують векторний додаток до скалярних x87 FPU, для одинарної точності і типів даних з подвійною точністю. SSE2 також пропонує цілий набір операцій для типів даних, починаючи від 8bit до 64bit. Це робить векторні можливості архітектури на одному рівні з передовими свого часу процесорами x86. Ці інструкції можуть бути також використані в 32-бітному режимі, що дозволяє покращення стандартів 32-бітних додатків 32-розрядної версії Windows 8. SSE3 інструкції та останні Streaming SIMD Extensions набори інструкцій не є стандартними особливостями архітектури.

No-Execute bit

Апаратний **NX-Bit** (**No eXecute Bit** в процесорах AMD), або **XD-Bit** (**Execute Disable Bit** в процесорах Intel) це технологія, яка запобігає можливості виконання даних як коду. Біт "NX" (біт 63 запису таблиці сторінок) дозволяє операційній системі вказати, які сторінки віртуального адресного простору може містити виконуваний код, а які ні. Спроба виконати код з сторінки, не поміченої " виконуваною" призведе до порушення доступу до пам'яті, подібно спробі записати на сторінку тільки для читання. Аналогічна функція була доступна на процесорах x86 з моменту 80286 в якості ознаки дескрипторів сегментів.

Сегментовані рішення вже давно вважаються застарілим режимом роботи, і всі нинішні операційні системи в силі обійти його, встановлюючи всі сегменти базової адреси в нуль (в реалізації 32 біт) і обмежитися розміром 4 ГБ. AMD був першим x86 сім'ї, який відмовився від режиму лінійної адресації.

Видалення старих функцій

Кілька «Системних функцій» архітектури x86 не використовуються в сучасних операційних системах і не доступні на AMD64 в довгому режимі (64-бітному і сумісному). Вони включають в себе сегментовану адресацію (хоча сегменти FS і GS зберігаються в залишковому вигляді для використання в якості додаткових базових вказівників в операційній системі), механізм task state switch, і віртуальний режим 8086 . Ці особливості залишаються повністю реалізовані в "стандартному режимі", що дозволяє процесорам працювати як з 32-розрядними, так і 16-бітовими операційними системами без змін. Деякі інструкції, які рідко використовуються, не підтримуються в 64-бітному режимі, у тому числі збереження / відновлення сегментних регістрів в стеку, збереження / відновлення всіх регістрів (PUSHA / POPA), десяткової арифметики, і "дальні "стрибки з безпосередніми операндами.

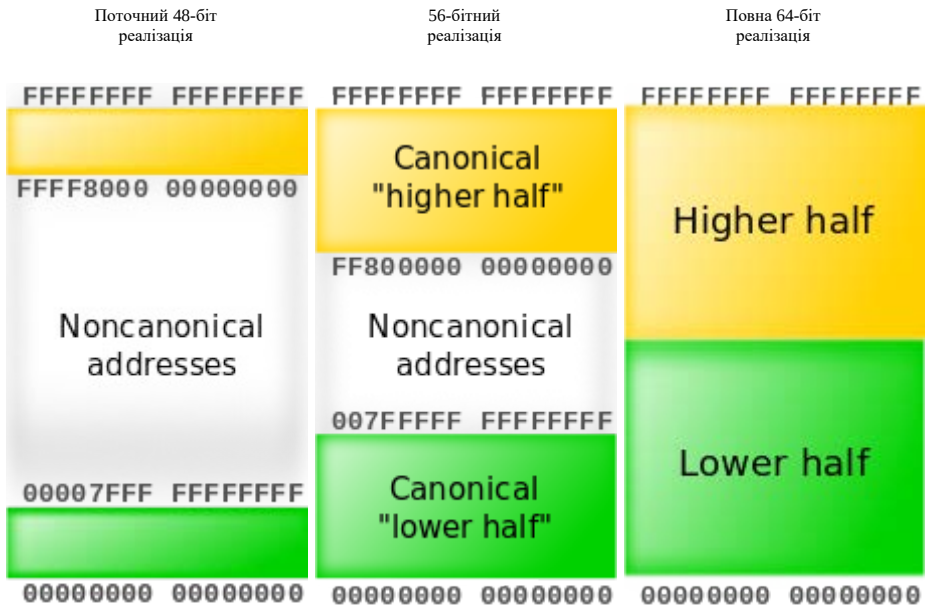
Віртуальний адресний простір (подробиці)

Хоча віртуальні адреси дорівнюють 64 біти в 64-бітному режимі, поточні реалізації не дозволяють використовувати весь віртуальний адресний простір з 2^{64} байт (16 EB). Це було б приблизно в чотири мільярди разів більше за розмір віртуального адресного простору на 32-розрядних машинах. Більшості операційних систем і додатків не потрібний такий великий адресний простір в осяжному майбутньому, тому реалізації такого широкого віртуального адресного простору просто збільшують складність і вартість трансляції адрес без реальної користі. Тому AMD вирішила, що в перших реалізаціях архітектури, тільки молодші 48 біт віртуальної адреси буде використано в трансляції адрес (в таблиці сторінок пошуку).

Крім того, специфікація AMD вимагає, щоб біти з 48 до 63 будь-якої віртуальної адреси повинні бути копіями біта 47, інакше процесор згенерує виняток. Адреси, сформовані за дотримання цього правила, називають "канонічною формою."

Ця функція полегшує масштабованість, правда 64-розрядної адресації. Багато операційних систем (у тому числі, і сім'я Windows NT) залишають верхню половину адресного простору (простір ядра) для себе і віддають нижню адресну половину (простір користувача) для коду програми, в режимі користувача . Стеки, купи та інші області даних гарантує дизайн "канонічних адрес", так що кожна сумісна реалізація AMD64 має, по суті, дві половинки пам'яті: нижня половина починається з 00000000'00000000 і "росте вгору", як більш віртуальна адреса. Крім того, дотримання «канонічної форми» адрес, перевіряючи невикористовувані біти адреси запобігає їх використання операційною системою в помічених вказівниками , як ознаки, привілеїв маркерів і т.д., а таке використання може стати проблематичним, коли архітектура має продовжений до реалізації більш віртуальний біт адреси.

Перші версії Windows, для 64 навіть не використовують повних 256 ТБ; вони були обмежені тільки 8 ТБ простору користувача і 8 ТБ простору ядра. ОС Windows не підтримує весь 48-розрядний адресний простір, у т.ч. Windows 8.1 (Windows 8 Professional - 512 GB)



Реалізації канонічного адресного простору (схема не в масштабі).

Структура таблиці Page

"Довгий режим " 64-розрядної адресації є надбудовою розширення фізичних адрес (PAE); через це розміри сторінки можуть бути 4 КВ або 2 МБ. Довгий режим також підтримує розміри сторінки з 1 ГБ (2^{30} байт). Система працює в довгому режимі , тому може використовувати чотири рівні таблиці сторінок, а не три рівні таблиці сторінок, як це є в режимі PAE: *Page-довідник таблиці вказівників* PAE продовжується від 4 записів до 512, а також додаткову у *Таблиці* додано *сторінку-Map Рівень 4 (PML4)*, що містять 512 записів в 48-бітових реалізаціях. У реалізаціях, що забезпечують великі віртуальні адреси, це останнє буде або зростати, щоб вмістити достатню кількість записів, щоб описати весь діапазон адрес, до теоретичного максимуму 33554432 записів для реалізації 64-розрядної або бути більш упорядкованим за нового рівня відображення, такі як PML5. Повне відображення ієрархії по сторінках по 4 Кбайта для всього 48-бітного простору займе трохи більше, ніж 512 ГБ оперативної пам'яті (близько 0,196% від 256 ТБ віртуального простору).

Обмеження Операційної системи

Операційна система може також обмежити віртуальний адресний простір.

Деталі фізичного адресного простору

Існуючі реалізації AMD64 підтримують фізичний адресний простір до 2^{48} байт оперативної пам'яті, або 256 ТБ. Верхня межа пам'яті, яка може бути використана в даній x86-64 системі, залежить від різних факторів, і може бути набагато менше, ніж та, що реалізована за допомогою процесора. Наприклад, станом на червень 2014 року, не було жодних відомих материнських плат для x86-64 процесорів, що підтримують 256 Тб оперативної пам'яті. (відомі 512 Гб)

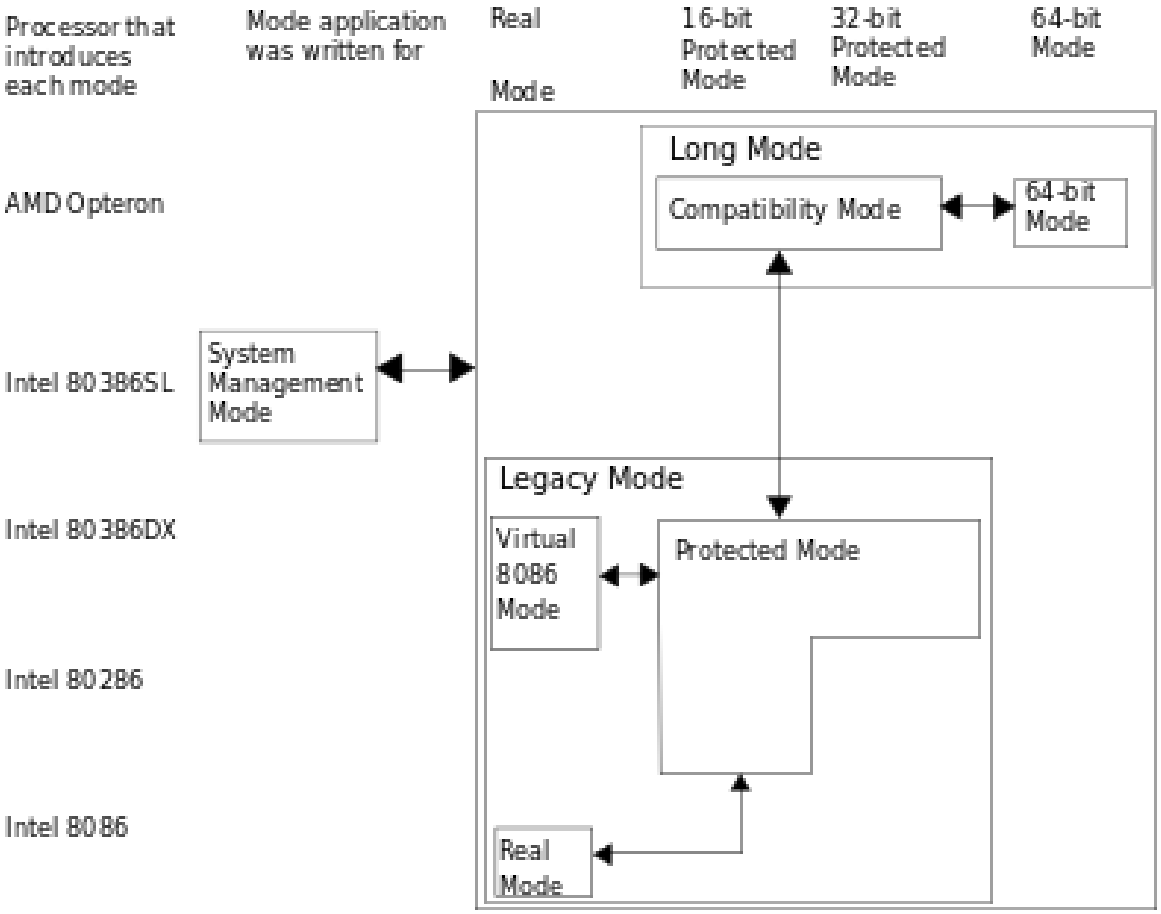
Операційна система може розмістити додаткові обмеження на обсяг оперативної пам'яті, що може використовуватися або не підтримується.

Режими роботи

Робочий режим	Операційний підрежим	Операційна система потрібна	Тип коду	Розмір адреси за замовчуванням	Розмір операнда за замовчуванням	Підтримувані типові розміри операндів	Регістр файл розмір	Типова GPR ширина
Довгий режим	64-бітний режим	64-розрядна операційна система або завантажувач	64-бітний код	64 біт	32 біт	8, 16, 32 або 64 біта	16 регістрів на файл	64 біт
	Режим сумісності	64-розрядна операційна система або завантажувач	32-бітний код захищеного режиму	32 біт	32 біт	8, 16 або 32 біта	8 регістрів на файл	32 біт
		64-розрядна операційна система	16-бітний код захищеного режиму	16 біт	16 біт	8, 16 або 32 біта	8 регістрів на файл	32 біт
Застарілий (спадковий) режим	Захищений режим	32-розрядна ОС або завантажувач, або 64-розрядний завантажувач	32-бітний код захищеного режиму	32 біт	32 біт	8, 16 або 32 біта	8 регістрів на файл	32 біт

		16-розрядний захищений режим ОС або завантажувач, або 32- або 64-бітний Завантажувач	16-бітний код захищеного режиму	16 біт	16 біт	8, 16 або 32 біта	8 регістрів на файл	16 або 32 біт
	Режим віртуального 8086	16- або 32-бітний захищений режим ОС	16-бітний код реального режиму	16 біт	16 біт	8, 16 або 32 біта	8 регістрів на файл	16 або 32 біт
	Реальний	16-бітному режимі операційної системи або завантажувач, або 32- або 64-бітний Завантажувач	16-бітний код реального режиму	16 біт	16 біт	8, 16 або 32 біта	8 регістрів на файл	16 або 32 біт

Традиційний стек регістрів FPU x87 не входить до складу розширення розміру регістрового файлу в 64-бітному режимі, у порівнянні з регістрами XMM, використовуваних SSE2, які дійсно ставали продовженим стеком регістрів X87 не просто регістрового файлу, хоча він дозволяє прямий доступ до окремих регістрів обмінних операцій. Також відзначимо, що 16-бітний код, написаний для 80286 і нижче, не можуть використовувати регістри 32-розрядні загального призначення (GPRS). 16-бітний код, написаний для 80386 і вище можна використовувати 32-бітові GPRS, але за замовчуванням використовує 16-бітові операнди.



Діаграма станів x86-64 режимів

Архітектура має два основні режими роботи, тривалий (довгий) і спадковий (Legacy).

Довгий режим (Long Mode)

Основний режим роботи - це поєднання режиму 64-бітного процесора і режиму комбінованої 32-бітної і 16-бітної сумісності. Він використовується 64-розрядними операційними системами. Під 64-бітною операційною системою 64-бітові програми працюють в 64-бітному режимі, і 32-бітові та 16-бітові додатки у захищеному режимі (які не повинні використовувати або реальний режим або віртуальний режим 8086) запускають в режимі сумісності. Програми та програми реального режиму, які використовують віртуальний режим 8086, не можуть працювати в тривалому режимі, якщо ці режими не емулюються програмно. Однак, такі програми можуть бути запущені з операційної системи, що працює в тривалому режимі на процесорах підтримка VT-X або AMD-V , створюючи віртуальний процесор, що працює в потрібному режимі.

У основному режимі майже немає втрати продуктивності для виконання захищеного коду режиму x86. Це на відміну від Intel IA-64 , де відмінності в наборі базових інструкцій означають, що для запуску 32-бітного коду він повинен бути виконаний або в режимі емуляції x86 (що робить процес повільнішим) або виділеного процесора x86. Тим не менше, на платформі x86-64 багато додатків x86 могли б виграти від 64-бітної перекомпіляції через додаткові регістри в 64-бітному коді і гарантованої SSE2 підтримки FPU, які компілятор може використовувати для оптимізації. Однак, для додатків, які повинні регулярно обробляти цілі числа ширші, ніж 32 біти, такі як криптографічні алгоритми, знадобиться переписати код обробки великих чисел для того, щоб скористатися 64-розрядними регістрами.

Спадковий режим (Legacy Mode)

Режим, який використовується у 16-бітовому («захищеному режимі» або «реальному режимі») 32-бітових операційних систем. У цьому режимі процесор працює як 32-бітний процесор x86, і тільки 16-розрядний і 32-бітний код може бути виконаний. Застарілий режим дозволяє використовувати максимум 32 біт віртуальної адресації, що обмежує віртуальний адресний простір 4 Гб. 64-розрядні програми не можуть бути запущені у стандартному режимі.

Реалізації AMD64

Архітектуру AMD64 реалізують наступні процесори:

- Opteron
- Athlon 64
- Athlon 64 X2
- Athlon 64 FX
- Athlon II (з подальшим розширенням "X2", "X3", або "X4", для вказання кількості ядер)
- Turion 64
- Turion 64 X2
- Sempron ("Палермо" E6 активізації і всі моделі "Маніла")
- Phenom (з подальшим "X3" або "X4")
- Phenom II (з подальшим "X2", "X3", "X4" або "x6")
- FX
- Fusion

Intel 64

Intel 64 є реалізація Intel після x86-64. Він використовується в більш нових версіях процесорів Pentium4 , Celeron , Celeron D , Xeon і Pentium Dual-Core, в Atom 230, 330, D410, D425, D510, D525, N450, N455, N470, N475, N550, N570, N2600 і N2800 і у всіх версіях Pentium Extreme Edition , Core 2 , Core i7 , Core i5 і Core i3 процесорів.

Історія Intel 64

Історично склалося так, що AMD розробляє і виробляє процесори з набором інструкцій за зразком оригінального дизайну Intel, але з x86-64, ролі помінялися: Intel виявилася в положенні прийняття архітектури команд (ISA), які створили AMD в якості доповнення до своєї лінійки процесорів x86 від Intel ,

Проект Intel був спочатку під кодовою назвою **Yamhill** (назва річки Ямхілл в штаті Орегон Willamette Valley). Після декількох років, заперечуючи його існування, Intel оголосила у лютому 2004 у IDF , що проект дійсно виконується. Голова ради директорів Intel в той час, Крейг Барретт , зізнався, що це був один з своїх найгірших збережених секретів.

Назва для цього набору команд змінювалася компанією Intel кілька разів. Протягом деякого часу почали називати її **IA-32e** (для IA-32 розширень) а в березні 2004 компанії Intel представила "офіційне" ім'я **EM64T** (Extended Memory 64 Technology). Наприкінці 2006 Intel почала використовувати замість імені **Intel 64** паралельно найменування AMD64.

Реалізації Intel 64

Першим у реалізації Intel 64 процесор був процесорний сокет Xeon під кодовою назвою *Nocona* в 2004 р. На відміну від цього, початкові фішки Прескотт (лютий 2004) не включити цю функцію. Згодом Intel почала продавати Intel 64 з підтримкою Pentium 4 за допомогою перегляду E0 ядра Prescott, продаються на ринку OEM як Pentium 4, модель F. E0 ревізія додає також Execute Disable (XD) (назва компанії Intel для біт NX). Офіційний запуск Intel з Intel 64 (під назвою EM64T в той час) для звичайних настільних процесорів була N0 Stepping Prescott-2M. Всі процесори 9xx, 8xx, 6xx, 5x9, 5x6, 5x1, 3x6, і 3x1 серії мають включену Intel 64, які в

майбутньому використають як процесори Intel для робочих станцій і серверів. Intel 64 також присутній в останніх членів лінійки Celeron D.

Перший Intel мобільний процесор реалізації Intel 64 Merom версія Core 2 процесора, був випущений 27 липня 2006. Жоден з раніше випущених процесорів Intel для нотбуків (Core Duo , Pentium M , Celeron M , Mobile Pentium 4) не реалізує Intel 64 .

Процесори реалізації архітектури Intel 64:

NetBurst мікроархітектури

- Xeon (всі моделі, такі як " Nocona ")
- Celeron (деякі моделі, такі як " Prescott ")
- Pentium 4 (в деяких моделях, такі як "Prescott")
- Pentium D
- Pentium Extreme Edition

Основні мікроархітектури

- Xeon (всі моделі типу "Woodcrest")
- Core 2 (у тому числі мобільних процесорів, такі як "Merom")
- Pentium Dual-Core (E2140, E2160, E2180, E2200, E2220, E5200, E5300, E5400, E6300, E6500, T2310, T2330, T2370, T2390, T3200 і T3400)
- Celeron (Celeron 4x0; Celeron M 5xx; E3200, E3300, E3400)

Мікроархітектура Atom

- 200 серія (не плутати з серією N200, яка широко використовується в нетбуках)
- 300 серія
- N4xx, серія N5xx
- Серія Dxxx

Nehalem , Sandy Bridge , Ivy Bridge і Haswell мікроархітектури

- Core i3
- Core i5
- Core i7

Реалізація x86-64 від VIA

VIA Technologies представила свій варіант архітектури x86-64 в 2008 році, після п'яти років розвитку своєї Centaur Technology . Під кодовою назвою "Ісая", 64-розрядна архітектура була представлена 24 січня 2008, а 29 травня під торговою маркою VIA Nano.

Процесор підтримує ряд VIA-певними розширеннями x86, призначених для підвищення ефективності в малопотужних приладах. Очікується, що архітектура Ісая буде мати продуктивність в два рази швидше в цілочисельних операціях і в чотири рази швидше в плаваючою комою, як попереднього покоління VIA Esther при еквівалентній тактовій частоті . Також очікується, що споживана потужність, буде на одному рівні з попереднім поколінням VIA процесорів, з Thermal Design Power від 5 Вт до 25 Вт.

Відмінності між AMD64 і Intel 64

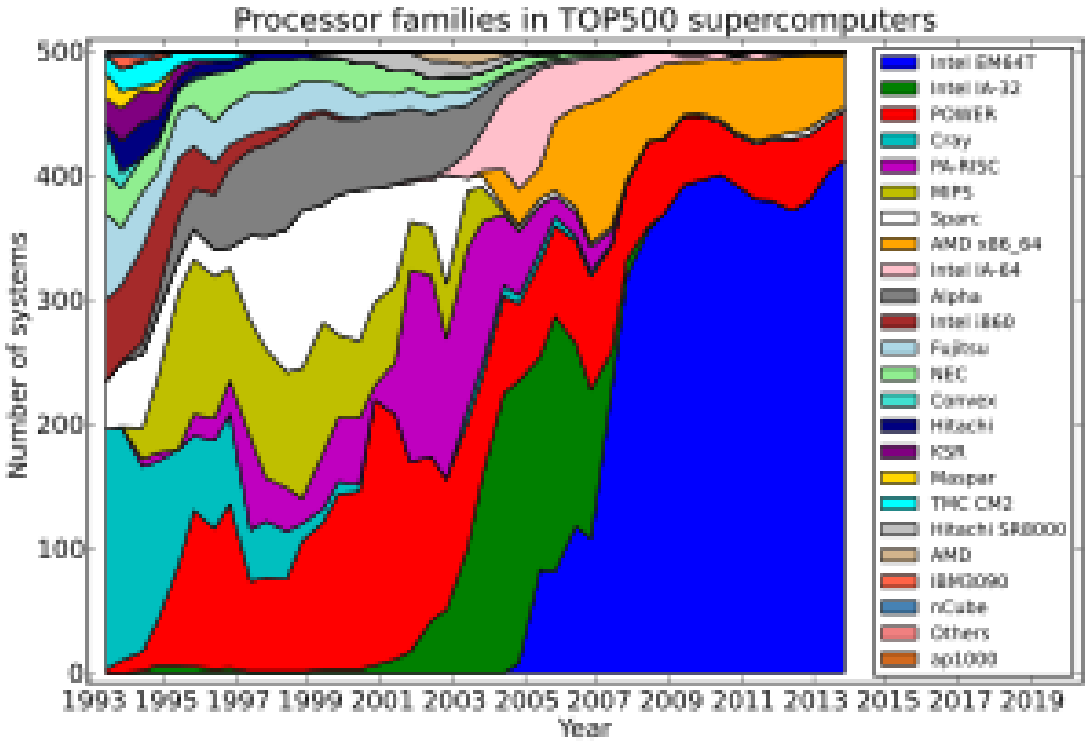
Хоча вони майже ідентичні, є деякі відмінності між цими двома наборами команд в семантиці кількох рідко використовуваних машинних інструкцій, які в основному використовуються для системного програмування . Компілятори зазвичай продукують виконувані файли (тобто машинний код), які уникають будь-яких відмінностей, принаймні для звичайних прикладних програм. Ці відмінності, становлять інтерес головним чином для розробників компіляторів, операційних систем і їм подібних, які повинні мати справу з індивідуальними і спеціальними інструкціями системи.

В останніх реалізаціях

- Intel 64 в BSF і BSR інструкціях (Bit Scan Forward) діє по-іншому, ніж AMD64-х, коли джерела дорівнюють нулю, і розмір операнда дорівнює 32 біт. Процесор встановлює ознаку нуля і залишає верхні 32 бітів призначення не визначеними.
- AMD64 має інший формат оновлення мікрокоду та контролю MSRs (модель-специфічних регістрів), тоді як Intel 64 знаряддя поновлення мікрокоду змінив тільки порівняно з 32-бітними процесорами.
- Intel 64 не вистачає деяких MSRs, які вважаються архітектурними в AMD64. Вони включають в себе `SYSCFG`, `TOP_MEM` і `TOP_MEM2`.
- Intel 64 дозволяє `SYSCALL` / `SYSRET` тільки в 64-бітному режимі (не в режимі сумісності), і дозволяє `SYSENTER` / `SYSEXIT` в обох режимах. AMD64 не вистачає `SYSENTER` / `SYSEXIT` в обох підрежимах тривалого режиму .
- У 64-бітному режимі префікс 66h (префікс розміру операнда) веде себе по-різному. Intel 64 ігнорує цей префікс, а AMD64 використовує 16-бітовий зсув поля в інструкції, і очищає верхні 48 біт вказівника інструкцій.
- Процесори AMD мають кращі можливості виконання команд з плаваючою комою. Невірний виняток при виконанні `FLD` або `FSTP` з 80-бітами сигналізує NaN, в той час як процесори Intel цього не мають.

- У Intel 64 відсутня можливість збереження і відновлення режиму з плаваючою комою (залучення FXSAVE iFXRST0R інструкцій).
- Останні процесори AMD64 знову мають обмежену підтримку сегментації, через довгий режим (LMSLE), щоб полегшити віртуалізацію для 64-розрядних гостей.

Реалізації



Площа діаграми показує предстаництво різних сімейств мікропроцесорів в списку рейтингу TOP500 суперкомп'ютерів, від 1992 до 2014.

Станом на 2014 переважно використовується не-x86 архітектура процесора, а досі в Суперкомп'ютерах Power Architecture використовується IBM POWER microprocessors (червоний колір на діаграмі), далеко позаду в чисельності на TOP500 є SPARC, в той час як недавно Суперкомп'ютер без співпроцесорів на основі Fujitsu SPARC64 VIIIfx досяг першої позиції, і все ще перебуває в першій десятці. Архітектура Non-CPU співпроцесорів (GPGPU) також зіграла велику роль в продуктивності. Intel, Xeon Phi співпроцесори, які реалізують підмножину x86-64 з деякими Vector Extensions, також використовуються, поряд з x86-64 процесорами в суперкомп'ютері Tianhe-2.

Сумісність з операційними системами і характеристики

Наступні операційні системи та релізи підтримують архітектуру x86-64 в тривалому режимі .

BSD

DragonFly BSD

Попередня робота з інфраструктурою була розпочата в лютому 2004 року на x86-64 порту. Цей розвиток пізніше зайшов у глухий кут. Розвиток продовжено знову протягом липня 2007 [50] і тривав протягом 2008 і 2009 року. Перший офіційний реліз, що містить x86-64 підтримку, була версія 2.4.

FreeBSD

FreeBSD вперше додав x86-64 підтримку під назвою "amd64" в якості експериментальної архітектури в 5,1-релізі, в червні 2003 року був включений в якості стандартної архітектури розподілу, 5,2-реліз, в січні 2004 року. З тих пір, FreeBSD призначив його в якості Рівня 1 платформи. В даний час ведеться робота по включенню більш повно x86 бінарного інтерфейсу додатків (ABI).

NetBSD

Вперше зроблено спробу підтримки x86-64 архітектури до NetBSD дерева вихідних текстів 19 червня 2001. NetBSD 2.0, випущена 9 грудня 2004, *NetBSD / amd64* це повністю інтегрована і підтримується порт. 32-бітний код і раніше підтримувався в 64-бітному режимі, а з NetBSD 32-розрядні ядра сумісні для 32-бітних системних викликів. NX біт використовується для забезпечення не виконуваних стека і купи з кожної сторінки деталізації (сегмент деталізації використовується в 32-бітній x86).

OpenBSD

OpenBSD підтримує AMD64 з OpenBSD 3.5, що вийшла 1 травня 2004. Розробники OpenBSD взяли до платформи підтримку біта NX , що дозволило легко здійснити W ^ X функції.

Код для порту AMD64 в OpenBSD також працює на Intel 64 процесорах, які містить клоноване використання розширень AMD64; У Intel 64 Процесорах додано NX під назвою "XD біт". Симетрична багатопроцесорна обробка (SMP) працює на порту AMD64 OpenBSD, починаючи з випуску 3.6 (2004).

DOS

Можна ввести довгий режим під DOS без подовжувача DOS, але користувач повинен повернутися в реальний режим для того, щоб викликати BIOS або DOS переривання.

Він також може бути можливим у довгому режимі з подовжувачем DOS , аналогічно DOS / 4GW , але більш складний, тому що x86-64 не вистачає віртуального режиму 8086 .

Linux

Linux був першим ядром операційної системи для запуску архітектури x86-64 в тривалому режимі , починаючи з версії 2.4 у 2001 році (до готовності фізичного апаратно). Linux також забезпечує зворотну сумісність для запуску 32-бітових виконуваних файлів. Це дозволяє програмам бути повторно в тривалому режимі при збереженні використання 32-бітових програм. Кілька дистрибутивів в даний час поставляються з x86-64 рідними ядрами і userlands. Деякі з них, такі як Arch Linux , SUSE , Mandriva , і Debian GNU / Linux , дозволяють користувачам встановити набір 32-бітових компонентів і бібліотек при установці з 64-розрядної DVD, що дозволяє більшості існуючих 32-розрядних додатків працювати разом з 64-бітною ОС. Інші дистрибутиви, такі як Fedora , Slackware і Ubuntu , доступні в одній версії, скомпільовані для 32-бітної архітектури, а у іншій складений для 64-бітної архітектури. Fedora і Red Hat Enterprise Linux дозволяють паралельні установки всіх компонентів користувацькими як в 32 так і 64-бітових версіях на 64-бітні системи.

x32 ABI (Application Binary Interface), введений в Linux 3.4, дозволяє компілювати програми для x32 ABI, для виконання в 64-розрядному режимі з x86-64 при цьому використовується тільки 32-бітові вказівники і поля даних. Це обмежує для програм віртуальний адресний простір до 4 ГБ, також зменшує пам'ять, займану програмою і в деяких випадках може дозволити йому працювати швидше. 64-бітний Linux дозволяє використовувати до 128 ТБ віртуального адресного простору для окремих процесів, і можете звернутися приблизно до 64 Тб фізичної пам'яті, з врахуванням обмежень процесора і системи.

OS X

Mac OS X 10.4.7 і більшість нових версій Mac OS X v10.4 запускають 64-бітові утиліти командного рядка, використовуючи POSIX (Portable Operating System Interface for uniX) і математичні бібліотеки на 64-розрядних машинах на базі Intel, як і всі версії Mac OS X v10.4 і 10,5 запускали їх на 64-бітних PowerPC машинах. Ніякі інші бібліотеки або рамки не працюють з 64-бітними додатками на Mac OS X v10.4.

Mac OS X v10.5 підтримує 64-розрядні додатки з графічним інтерфейсом при використанні Cocoa, Quartz , OpenGL і X11 на 64-розрядних машинах на базі Intel, а також на 64-бітних PowerPC машин. Всі бібліотеки не-GUI також підтримують 64-розрядні додатки на цих платформах. Ядро і всі розширення ядра, 32-розрядна версія.

Mac OS X 10.6 є перший варіант OS X, який підтримує 64-бітове ядро. Однак, не всі 64-бітові комп'ютери можуть запускати 64-бітове ядро, і не всі 64-бітові комп'ютери, які можуть запустити 64-бітове ядро будуть робити це за замовчуванням. 64-бітове ядро, як 32- бітове ядро, підтримує 32-розрядні додатки; обидва ядра підтримують 64-розрядні додатки. 32-розрядні додатки мають обмеження віртуального адресного простору в 4 ГБ під будь ядром.

OS X v10.8 включає в себе тільки 64-бітове ядро, але як і раніше підтримує 32-розрядні додатки.

OS X використовує універсальний бінарний формат для упаковки 32- і 64-розрядних версій додатків і бібліотек в одному файлі; найбільш підходящий варіант вибирається автоматично під час завантаження. У Mac OS X 10.6, універсальний двійковий формат також використовується для ядра і для тих розширень ядра, які підтримують як 32-розрядні, так і 64-бітові ядра.

Solaris

Solaris 10 і пізніших версій підтримує архітектуру x86-64.

Для ОС Solaris 10, так само, як у SPARC архітектурі, є тільки образ системи з експлуатації, в якому міститься 32-розрядне ядро і 64-розрядне ядро; це помічені як "64 / x86" DVD-ROM зображення. За замовчуванням

завантажується 64-бітове ядро, що дозволяє для запуску як 64-розрядні і існуючі або нові 32-бітові виконувані файли. 32-бітове ядро також може бути обране вручну, і в цьому випадку будуть працювати тільки 32-бітові виконувані файли. `ISAInfo` команда може бути використана для визначення, якщо система працює в 64-бітовому ядрі.

Для ОС Solaris 11 надається тільки 64-бітове ядро . Тим не менш, 64-бітове ядро підтримує як 32, так і 64-розрядні виконувані файли, бібліотеки та системні виклики.

Windows

64-розрядні випуски клієнта Microsoft Windows і server- Windows XP Professional x64 Edition і Windows Server 2003 x64 Edition-випущені в березні 2005 року. Windows Vista , яка також має багато різних видань, була випущена в січні 2007 року. Windows 7 була випущена в липні 2009 року . Windows Server 2008 R2 і пізніші версії доступна в якості тільки.x64 версії .

До Windows 8.1 / Windows Server 2012 R2, Windows для x64 пропонується:

- 8 ТБ віртуального адресного простору в процесі, який доступний як з користувальницького режиму так і режиму ядра. Програма x64 можна використовувати все це, за умови резервної межі памяті в системі, і за умови, що задано опцію «large address aware ". Це 4096-кратне збільшення в порівнянні з 2 ГБ в режимі користувача за замовчуванням віртуального адресного простору на 32-бітних Windows.
- 8 ТБ режиму ядра віртуального адресного простору для операційної системи. Як і в режимі користувача адресного простору, це збільшення в 4096 разів більше 32-розрядних версій Windows. Більший простір, насамперед, на користь кешу файлової системи і ядра режиму "купи" (невивантажуваного пулу і переміщуваного пулу). Windows використовує тільки 16 ТБ з 256 ТБ, здійснюваної процесорів, адже ранні процесори AMD64 не вистачало `CMPXCHG16B` інструкції. [70]
- Під Windows 8.1 і Windows Server 2012 R2, як у режимі користувача і у режимі ядра віртуальний адресний простір розширені до 128 ТБ. Ці версії Windows, не встановлюватимуть на процесорах, які не містять `CMPXCHG16B` інструкцій.

Додаткові характеристики застосовні до всіх 64-розрядних версій Windows, наступні:

- Можливість запускати існуючі 32-розрядні додатки (`.exe` програм) і бібліотеки Dynamic Link (`.dll`) за допомогою WoW64 . Крім того, 32-бітна програма, якщо це було пов'язано з опцією " в курсі великої адреси ", можна використовувати до 4 ГБ віртуального адресного простору в 64-бітних ОС Windows, замість використовуваного за замовчуванням 2 ГБ (опціонально 3 ГБ з `/ 3GB` опції завантаження і опції " large address aware "), пропонованої для 32-бітної ОС Windows. На відміну від використання `/ 3GB` варіанту завантаження на x86, це не зменшує в режимі ядра віртуальний адресний простір, доступний для операційної Системи. Тому 32-бітові додатки можуть отримати вигоду з роботи на x64 ОС Windows, навіть якщо вони не перекомпільовані для x86-64.
- 32- і 64-розрядні додатки, якщо не пов'язані з "великою адресою," обмежуються 2 ГБ віртуального адресного простору.
- Можливість використовувати до 128 Гб (Windows XP / Vista), 192 Гб (Windows 7), 512 Гб (Windows 8), 1 ТБ (Windows Server 2003), 2 ТБ (Windows Server 2008) або 4 ТБ (Windows Server 2012) фізичної пам'яті (ОЗУ).
- LLP64 модель даних: "INT" і "довгі" типи 32 біт, тепер становить 64 біта, у той час як вказівники і типи, отримані з вказівників є 64-розрядними.
- Драйвери пристрою режиму роботи ядра повинні бути 64-розрядної версії; немає ніякого способу, щоб запустити 32-розрядні виконувані файли режиму ядра в межах 64-розрядної операційної системи. Драйвери пристрою режиму Системи можуть бути або 32-розрядні або 64-розрядні.
- 16-бітові Windows (Win16) і DOS програми не будуть працювати на x86-64 версіях Windows, через вилучення віртуальної DOS машини підсистемою (NTVDM), яка спирається на здатність використовувати віртуальний режим 8086. Режим віртуального 8086 не може бути введений під час роботи в тривалому режимі.
- Повна реалізація NX (No Execute) Функції захисту сторінки. Це здійснюється також у останніх 32-розрядних версіях Windows, коли вони почали працювати в режимі PAE.

- Замість FS дескриптора сегмента на x86 версіях сім'ї Windows NT, використовується GS дескриптор сегмента, щоб вказати на дві операційні системи, що визначають структури: Thread Information Block (NT_TIB) в режимі користувача і процесор управління області (KPCR) в режимі ядра. Так, наприклад, в режимі користувача `GS:0` є адресою першого елемента інформаційного блоку.
- Найнедавніша документація від Microsoft заявляє, що x87 / MMX / 3DNow! Інструкції можуть бути використані в тривалому режимі, але вони є застарілими і можуть викликати проблеми сумісності в майбутньому.
- Деякі компоненти, такі як Microsoft Jet Database Engine і об'єктів доступу до даних НЕ БУДЕ портована на 64-бітні архітектури, такі як x86-64 і IA-64.
- Microsoft Visual Studio може компілювати власні додатки на цільові архітектури або архітектуру x86-64, які можуть працювати тільки на 64-бітній Microsoft Windows, або IA-32 архітектурі, яка може працювати як 32-розрядний додаток на 32-бітній ОС Microsoft Windows або 64-бітній Microsoft Windows в режимі емуляції WoW64. Керовані додатки можуть бути скомпільовані або в IA-32, x86-64 або в режимі AnyCPU. Програмне забезпечення, створене в перших двох режимах поводиться як їх IA-32 або x86-64 нативний код відповідно; однак при використанні режиму AnyCPU, застосування в 32-розрядних версіях Microsoft Windows працюють як 32-розрядні додатки, в той час як в 64-розрядних версіях Microsoft Windows вони працюють, як 64-розрядні додатки.

Відео ігрових консолей

PlayStation 4 і Xbox One включають Jaguar , як багатоядерний процесор , розроблений AMD. Обидва використовують x86-64 для використання 8 ГБ оперативної пам'яті.

Список використаних джерел From Wikipedia, the free encyclopedia

1. AMD Corporation (September 2012)."Volume 2: System Programming" (PDF). *AMD64 Architecture Programmer's Manual*. AMD Corporation. Retrieved 2014-02-17.
2. IBM Corporation (2007-09-06). "IBM WebSphere Application Server 64-bit Performance Demystified". p. 14. Retrieved 2010-04-09.Figures 5, 6 and 7 also show the 32-bit version of WAS runs applications at full native hardware performance on the POWER and x86-64 platforms. Unlike some 64-bit processor architectures, the POWER and x86-64 hardware does not emulate 32-bit mode. Therefore applications that do not benefit from 64-bit features can run with full performance on the 32-bit version of WebSphere running on the above mentioned 64-bit platforms.
3. "Debian AMD64 FAQ". *Debian Wiki*. Retrieved 2012-05-03.
4. "x86-64 Code Model". Apple. Retrieved November 23, 2012.
5. `arch(1)` – Darwin and Mac OS X General CommandsManual
6. Kevin Van Vechten (2006-08-09). "re: Intel XNU bug report".*Darwin-dev mailing list*. Apple Computer. Retrieved 2006-10-05. The kernel and developer tools have standardized on "x86_64" for the name of the Mach-O architecture
7. "Solaris 10 on AMD Opteron". Oracle. Retrieved 2010-12-09.
8. "Microsoft 64-Bit Computing". Microsoft. Retrieved 2010-12-09.
9. "AMD64 Port". Debian. Retrieved November 23, 2012.
10. "Gentoo/AMD64 Project". Gentoo Project. Retrieved May 27,2013.
11. "AMD Discloses New Technologies At Microporcessor Forum"(Press release). AMD. 1999-10-05. Retrieved 2010-11-09.
12. "AMD Releases x86-64 Architectural Specification; Enables Market Driven Migration to 64-Bit Computing" (Press release). AMD. 2000-08-10. Retrieved 2010-11-09.
13. "Intel 64 and IA-32 Architectures Software Developer's Manual, Volume 3A: System Programming Guide, Part 1" (PDF). pp. 4–10. Retrieved 2010-05-29.
14. "BIOS and Kernel Developer's Guide (BKDG) For AMD Family 10h Processors" (PDF). p. 24. Retrieved 2010-05-29. Physical address space increased to 48 bits.[*dead link*]
15. "Myth and facts about 64-bit Linux" (PDF). 2008-03-02. p. 7. Retrieved 2010-05-30. Physical address space increased to 48 bits
16. Shanley, Tom (1998). *Pentium Pro and Pentium II System Architecture*. PC System Architecture Series (Second ed.). Addison-Wesley. p. 445. ISBN 0-201-30973-4.
17. Microsoft Corporation. "What is PAE, NX, and SSE2 and why does my PC need to support them to run Windows 8 ?". Retrieved19 March 2013.

18. "Memory Limits for Windows Releases". *MSDN*. Microsoft. November 16, 2013. Retrieved January 20, 2014.
19. Jeff Tyson (2004-01-15). "How Virtual Memory Works". howstuffworks. Retrieved 2010-06-07. The read/write speed of a hard drive is much slower than RAM, and the technology of a hard drive is not geared toward accessing small pieces of data at a time. If your system has to rely too heavily on virtual memory, you will notice a significant performance drop. The key is to have enough RAM to handle everything you tend to work on simultaneously.
20. Jason Dunn (2003-09-03). "Computer RAM: A Crucial Component in Video Editing". Microsoft Corporation. Retrieved 2010-06-09. There's a point of diminishing returns where adding more RAM won't give you better system performance.
21. "Understanding Memory Configurations and Exchange Performance". Microsoft Corporation. 2003-09-03. Retrieved 2010.There is, however, a point of diminishing returns at which adding memory to the server may not be justifiable based on price and performance.
22. "Opteron 6100 Series Motherboards". Supermicro Corporation. Retrieved 2010-06-22.
23. "Supermicro XeonSolutions". Supermicro Corporation. Retrieved2010-06-20.
24. "Opteron 8000 Series Motherboards". Supermicro Corporation. Retrieved 2010-06-20.
25. "Tyan Product Matrix". MiTEC International Corporation. Retrieved2010-06-21.
26. "Craig Barrett confirms 64 bit address extensions for Xeon. And Prescott", from The Inquirer
27. "A Roundup of 64-Bit Computing", from internetnews.com
28. "Intel 64 Architecture". Intel. Retrieved 2007-06-29.
29. "VIA to launch new processor architecture in 1Q08" (subscription required). DigiTimes. Retrieved 2007-07-25.
30. Stokes, Jon (2008-01-23). "Isaiah revealed: VIA's new low-power architecture". Ars Technica. Retrieved 2008-01-24.
31. "VIA Launches VIA Nano Processor Family" (Press release). VIA. 2008-05-29. Retrieved 2008-05-29.
32. "VIA Isaiah Architecture Introduction" (PDF). VIA. 2008-01-23. Retrieved 2013-07-31.
33. Wasson, Scott. "64-bit computing in theory and practice". *The Tech Report*. The Tech Report. Retrieved 2011-03-22.
34. "Intel 64 and IA-32 Architectures Software Developer's Manual Volume 2 (2A, 2B & 2C): Instruction Set Reference, A-Z". Intel. September 2013. pp. 4–397. Retrieved 2014-01-21.
35. "Intel 64 and IA-32 Architectures Software Developer's Manual Volume 2 (2A, 2B & 2C): Instruction Set Reference, A-Z". Intel. September 2013. pp. 4–400. Retrieved 2014-01-21.
36. "AMD64 Architecture Programmer's Manual Volume 2: System Programming". AMD. May 2013. p. 33. Retrieved 2014-01-21.
37. "How retiring segmentation in AMD64 long mode broke VMware". Pagetable.com. 2006-11-09. Retrieved 2010-05-02.
38. "VMware and CPU Virtualization Technology" (PDF). VMWare. Retrieved 2010-09-08.
39. Maged M. Michael. "Practical Lock-Free and Wait-Free LL/SC/VL Implementations Using 64-Bit CAS". IBM. Retrieved 2014-01-21.
40. "http://blogs.msdn.com/b/oldnewthing/archive/2006/11/22/1122581.aspx" The Old New Thing, by Raymond Chen (see bottom paragraph of blog post)
41. "System Requirements—Windows 8.1". Retrieved 2014-04-27.To install a 64-bit OS on a 64-bit PC, your processor needs to support CMPXCHG16b, PrefetchW, and LAHF/SAHF.

42. "Revision Guide for AMD Athlon 64 and AMD Opteron Processors", from AMD
43. "AMD Turion 64 pictured up and running", from The Inquirer
44. "Athlon 64 revision E won't work on some Nforce 3/4 boards", from The Inquirer
45. "Intel 64 architecture increases the linear address space for software to 64 bits and supports physical address space up to 46 bits." on page Vol. 1 2-21 of Intel 64 and IA-32 Architectures Software Developer's Manual September 2014
46. "Statistics | TOP500 Supercomputer Sites". Top500.org. Retrieved2014-03-22.
47. "Intel® Xeon PhiTM Coprocessor Instruction Set Architecture Reference Manual". Intel. September 7, 2012. section B.2 Intel Xeon Phi coprocessor 64 bit Mode Limitations.
48. "Intel Powers the World's Fastest Supercomputer, Reveals New and Future High Performance Computing Technologies". RetrievedJune 21, 2013.
49. "cvs commit: src/sys/amd64/amd64 genassym.c src/sys/amd64/include asm.h atomic.h bootinfo.h coredump.h cpufunc.h elf.h endian.h exec.h float.h fpu.h frame.h globaldata.h ieeefp.h limits.h lock.h md_var.h param.h pcb.h pcb_ext.h pmap.h proc.h profile.h psl.h ...". Retrieved 2009-05-03.
50. "AMD64 port". Retrieved 2009-05-03.
51. "DragonFlyBSD: GoogleSoC2008". Retrieved 2009-05-03.
52. "Summer of Code accepted students". Retrieved 2009-05-03.
53. "DragonFlyBSD: release24". Retrieved 2009-05-03.
54. Tutorial for entering protected and long mode from DOS
55. Andi Kleen (2001-06-26). "Porting Linux to x86-64". Status: The kernel, compiler, tool chain work. The kernel boots and work on simulator and is used for porting of userland and running programs
56. Andi Kleen. "Andi Kleen's Page". This was the original paper describing the Linux x86-64 kernel port back when x86-64 was only available on simulators.
57. "Arch64 FAQ". 2012-04-23. You can either use the multilib packages or a i686 chroot.
58. Thorsten Leemhuis (2011-09-13). "Kernel Log: x32 ABI gets around 64-bit drawbacks". www.h-online.com. Archived from the original on October 28, 2011. Retrieved 2011-11-01.
59. "x32 – a native 32-bit ABI for x86-64". linuxplumbersconf.org. Retrieved 2011-11-01.
60. "x32-abi". Google Sites. Retrieved 2011-11-01.
61. "AMD64 Port". debian.org. Retrieved 2011-10-29.
62. "Apple – Mac OS X Xcode 2.4 Release Notes: Compiler Tools".Apple Inc. 2007-04-11. Archived from the original on 2009-04-22. Retrieved 2012-11-19.
63. "Apple – Mac OS X Leopard – Technology – 64-bit". Apple Inc. Archived from the original on 2009-01-12. Retrieved 2012-11-19.
64. "Mac OS X v10.6: Macs that use the 64-bit kernel". Apple Inc. Retrieved 2012-11-29.
65. John Siracusa. "Mac OS X 10.6 Snow Leopard: the Ars Technica review". Ars Technica LLC. Retrieved 2010-06-20.

66. "Mac OS X Technology". Apple Inc. Archived from the original on 2011-03-28. Retrieved 2012-11-19.
67. "/LARGEADDRESSAWARE (Handle Large Addresses)". *Visual Studio 2005 Documentation – Visual C++ – Linker Options*. Microsoft. Retrieved 2010-06-19. The /LARGEADDRESSAWARE option tells the linker that the application can handle addresses larger than 2 gigabytes.
68. Matt Pietrek (May 2006). "Everything You Need To Know To Start Programming 64-Bit Windows Systems". Microsoft. Retrieved2010-05-24.
69. Chris St. Amand (January 2006). "Making the Move to x64". Microsoft. Retrieved 2010-05-24.
70. "Behind Windows x86-64's 44-bit Virtual Memory Addressing Limit". Retrieved 2009-07-02.
71. ^ Jump up to:[a] [b] "64-bit programming for Game Developers". RetrievedAugust 21, 2013.
72. "Memory Limits for Windows Releases". Microsoft. RetrievedFebruary 20, 2013.
73. Microsoft Developer Network – General Porting Guidelines (64-bit Windows Programming)
74. Microsoft Developer Network – Data Access Road Map
75. Anand Lal Shimpi (2013-05-21). "The Xbox One: Hardware Analysis & Comparison to PlayStation 4". Anandtech. Retrieved 2013-05-22.
76. "The Tech Spec Test: Xbox One Vs. PlayStation 4". Game Informer. 2013-05-21. Retrieved 2013-05-22.
77. "An example file from Linux 3.7.8 kernel source tree displaying the usage of the term x86_64". Retrieved 2013-02-17.
78. ProcessorArchitecture Fields
79. "AMD plays antitrust poker for Intel's X86 licence". Incisive Media Limited. 2009-02-03. Retrieved 2009-02-26.
80. "Patent Cross License Agreement Between AMD and Intel". 2001-01-01. Retrieved 2009-08-23.
81. "AMD Intel Settlement Agreement".
82. Stephen Shankland and Jonathan E. Skillings (2009-11-12). "Intel to pay AMD $1.25 billion in antitrust settlement". CNET. Retrieved2012-04-24.