Lesson 1. What 64-bit systems are

* [Intel 64 architecture](http://www.viva64.com/en/l/0001/#ID0EGE)
* [64-bit operating systems](http://www.viva64.com/en/l/0001/#ID0EGAAC)
* [Address space](http://www.viva64.com/en/l/0001/#ID0ECBAC)
* [Win64 program model](http://www.viva64.com/en/l/0001/#ID0ENFAC)

By the moment of writing the course, there are two most popular [64-bit](http://www.viva64.com/en/t/0001/) architectures of microprocessors: IA64 and Intel 64.

1. **IA-64** is a 64-bit microprocessor architecture developed by Intel and Hewlett Packard companies together. It is implemented in Itanium and Itanium 2 microprocessors. To learn more about the architecture IA-64 see the following Wikipedia article "[Itanium](http://en.wikipedia.org/wiki/Itanium)".
2. **Intel 64** (EM64T / AMD64 **/** x86-64 / x64) is an extension of [x86](http://www.viva64.com/en/t/0060/) architecture with full backward compatibility. There are many variants of its name and it causes some confusion, but all these names mean the same thing: x86-64, AA-64, Hammer Architecture, AMD64, Yamhill Technology, EM64T, IA-32e, Intel 64, x64. To learn how so many names appeared see the article in Wikipedia: "[X86-64](http://en.wikipedia.org/wiki/X86-64)".

You should understand that IA-64 and Intel 64 are absolutely different, incompatible with each other, microprocessor architectures. Within the scope of this course we will consider only Intel 64 (x64 / AMD64) architecture as the most popular among applied Windows software developers. Accordingly, when we mention Windows operating system, we will mean its 64-bit versions for Intel 64 architecture. For example: Windows XP Professional x64 Edition, Windows Vista x64, Windows 7 x64. The program model Intel 64 available to a programmer in a 64-bit Windows is called [Win64](http://www.viva64.com/en/t/0055/), for short.

Intel 64 architecture

The information given here is based on the first volume of the documentation "[AMD64 Architecture Programmer's Manual. Volume 1. Application Programming](http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/24592.pdf)".

The architecture [Intel 64](http://www.viva64.com/en/t/0022/) we are considering here, is a simple yet powerful extension of the obsolete commercial architecture x86 with backward compatibility. It adds the 64-bit address space and extends resources to support higher performance of recompiled 64-bit programs. The architecture supports obsolete 16-bit and 32-bit code of applications and operating systems without modifying or recompiling them.

The need of a 64-bit architecture is determined by the applications that require a larger address space. First of all, these are high-performance servers, data managers, CAD and, of course, games. These applications will get great benefits from the 64-bit address space and larger number of registers. Few registers available in the obsolete x86 architecture limit performance of computing tasks. The increased number of registers provides the necessary performance for many applications.

Let us point out the main advantages of the architecture x86-64:

* the 64-bit address space;
* an extended register set;
* a command set familiar to developers;
* the capability to launch obsolete 32-bit applications in a 64-bit operating system;
* the capability to use 32-bit operating systems.

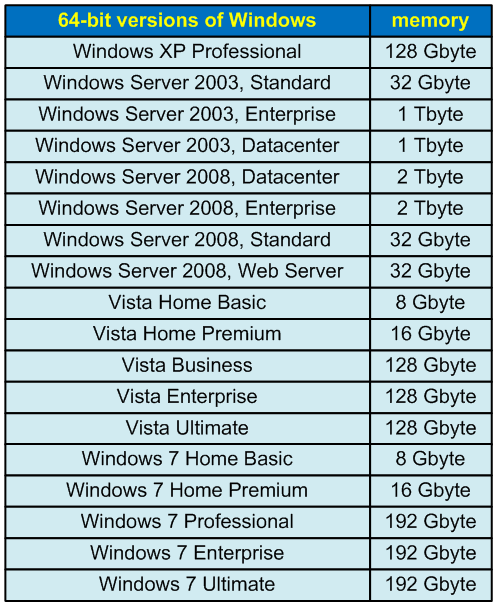
64-bit operating systems

Nearly all modern operating systems have versions for Intel 64 architecture. For example, Microsoft ships Windows XP x64. Large UNIX developers also ship 64-bit versions, for example, Linux Debian 3.5 x86-64. But it does not mean that the whole code of such a system is 64-bit. Some parts of the operating system and many applications may well remain 32-bit because Intel 64 provides backward compatibility. Thus, the 64-bit version of Windows uses a special mode [WoW64](http://www.viva64.com/en/t/0056/) (Windows-on-Windows 64) that translates the calls of 32-bit applications to the resources of the 64-bit operating system.

Address space

Although a 64-bit processor can theoretically address 16 Ebytes of memory (2^64), Win64 now supports only 16 Tbytes (2^44). There are some reasons for that. Contemporary processors can provide access only to one Tbyte (2^40) of physical memory. The architecture (but not the hardware part) can extend this space up to 4 Pbytes (2^52). But in this case you need an immense amount of memory for the page tables representing it.

Besides the limitations described above, the size of memory available in every particular version of the 64-bit Windows depends upon the commercial reasons of Microsoft. Different Windows versions have different limitations which are illustrated in the table.

[](http://www.viva64.com/media/images/content/l/01_What_64-bit_systems_are/01-01-en-big.png)

*Table 1 - The amounts of memory supported in different Windows versions*

Win64 program model

Like in [Win32](http://www.viva64.com/en/t/0054/), the size of a page in Win64 is 4 Kbytes. The first 64 Kbytes of the address space are never displayed, so the lowest correct address is 0x10000. Unlike Win32, system DLL's take more than 4 Gbytes.

Compilers for Intel 64 have one peculiarity: they can **use registers with great efficiency to pass parameters into functions instead of using the stack**. This allowed the Win64 architecture developers to get rid of such a notion as a [calling convention](http://en.wikipedia.org/wiki/Calling_convention). In Win32, you may use various conventions: \_\_stdcall, \_\_cdecl, \_\_fastcall, etc. In Win64, there is only one calling convention. Here is an example of how four arguments of *integer* type are passed through registers:

* RCX: the first argument
* RDX: the second argument
* R8: the third argument
* R9: the fourth argument

The arguments following the first four integer ones are passed through the stack. To pass float arguments XMM0-XMM3 registers are used as well as the stack.

The difference in the calling conventions makes it impossible to use both 64-bit and 32-bit code in one program. In other words, if an application has been compiled for the 64-bit mode, all the libraries (DLL) being used must also be 64-bit.

Passing parameters through registers is one of the innovations that make 64-bit programs faster than 32-bit ones. You may achieve an additional performance gain using 64-bit data types. We will tell you about it in the next lesson.