Origins and history

The IBM i Operating System (IBM i) is unique in the IT industry for its level of integration and for the underlying architecture of the operating system itself. Evolving, as it has, from the forward-thinking System/38 and AS/400 architectures, IBM i offers tremendous value in low total cost of ownership. It provides IT professionals a platform to implement and run business solutions while protecting the significant investments they have made over the years in mission-critical software.

The core architectures of IBM i were first created for the System/38. The S/38 architecture included the following:

- use of a relational database as the inherent file system;
- use of single-level storage to remove the necessity of customers managing storage;
- object-based operating system to ensure integrity and reliability;
- technology independent machine interface which allowed compiled programs to move from one release and one processor family to another without recompilation; and
- expectations that multiple workloads would be run on the machine and therefore required isolation and load-balancing capabilities.

The industrial strength architecture of the S/38 was unified with ease of use and management of its sister product, the System/36, when the AS/400 was introduced in 1988. Over the years, more capabilities were added to the operating system as it became the core of the iSeries and System i. Ultimately, the hardware and I/O platform of the AS/400 line converged with the hardware and I/O of the AIX-based RS/6000 until, in 2007, IBM Power Systems was launched. In early 2008, the operating system was given its current name, IBM i.

The latest version of the IBM i operating system is IBM i 7.3.

Integration is the key – inclusive middleware

From a customer point of view, the single defining characteristic of IBM i is its integration. In fact, the “i” in IBM i stands for “integration” and for good reason. There are two important aspects of the integration of IBM i: the core middleware components required by business applications and ensuring all of these components are designed to fit together and follow the IBM i architecture. This integration, along with built-in automation of many system and data management functions, allows customers to run their businesses, not their systems.

IBM i boasts an extensive list of key software features. These features have to be added on at an additional cost, or loaded and managed separately, or both, on other operating systems. These include:

- **Database**: The DB2 for i database is not merely packaged with IBM i. It is a part of the operating system. It is an SQL-compliant relational database, supporting the standards that are implied with that statement. DB2 for i does not merely make use of the file system on the platform, but is part of the file system, allowing for cooperative performance algorithms and easier tuning. DB2 for i is characterized by its level of automation, which allows many customers to have a minimal number of technical staff devoted to database management and administration. In fact, it is quite typical to have IBM i customers who have no database administrator at all, because of the automated optimization that is part of DB2 for i.
- **Multiple workload virtualization**: IBM i is designed to expect more than one workload to be run at once. There are a host of security capabilities which make isolation of data between workloads straightforward and secure, along with a subsystem design that allows customers to balance the importance of various workloads to ensure that each is getting the amount of system resources appropriate for its function. This is accomplished without needing to purchase additional virtualization or management software.
- **Web server**: IBM i ships the open source Apache Web Server and has ensured that it is connected to the IBM i security and file system models. IBM i keeps it up-to-date, including support for business-critical extensions such as the Payment Card Industry standards for using credit and debit cards within browser-based applications.
- **Web application server**: Similar to the open source Tomcat, the integrated web application server supports running Java 5 and Java 6 applications, which have significant capabilities beyond what traditional HTML-based applications can deliver.
- **Programming languages**: From traditional languages such as CL and RPG, to C/C++ and Java, to the most commonly used language for web programming, PHP, IBM i ships the underlying support for all of these languages as part of the base offering, with enhancements in every release. This language support is integrated with the software development tools created by IBM Rational and Zend Corporation.
- **High-availability and integrity underpinnings**: The ability to build applications which can reliably roll back to previous states is supported by DB2 for i, and also by the journaling capability inherent and integrated in IBM i. This allows for a simpler, more complete approach to highly available applications and data integrity than can be built on the standard file systems of other operating environments.
- **System management**: The IBM Systems Director Navigator is a browser-based management tool for IBM i which allows a user, operator, or administrator to control work on an IBM i partition. This tool is also available with no additional charge, and includes several

very powerful features, including a Performance Data Investigator that allows customers to track, monitor and plan for the performance and resource use on their system.

- **Additional capabilities** provided by IBM i as part of its integrated platform are listed below:
    - Security on IBM i
    - Encryption capabilities
    - Integrated web services for i
    - Digital certificate management
    - Integrated version of IBM Tivoli Director Server
    - Integrated connection to and management of x86 systems
    - Networking and network management

Architecture for simplicity, stability, and security

Another key differentiator for IBM i is the underlying architecture of the operating system which provides for simplicity, stability, and security. These are not add-on features but inherent aspects of the operating system. This section describes a few examples.

- **Simplicity**
  As mentioned above, the relational database in IBM i is an integrated component, and it is built on a set of integrated storage management unique to IBM i and its predecessors. This storage management is based on the single-level storage architecture that treats all of the storage managed by IBM i as if it is one long stream of memory that encompasses the system memory, as well as the storage on disks. This architecture makes it critical that the operating system decide where to store any given piece of data, and that in turn removes the necessity of the user to manage data placement. This makes management of storage, and the data base objects contained in that storage, significantly simpler.
- **Stability**
  Several architectural attributes of IBM i contribute to the stability of the system over the years. First, applications compiled on IBM i are compiled to a set of intermediate code instructions defined as the Technology Independent Machine Interface (TIMI). This allows IBM i to fundamentally change the implementation of underlying hardware, firmware and virtualization features, without requiring rewriting, changing, or even recompiling the applications written by users. This sort of forward compatibility is a tremendous business advantage and has allowed the movement of software developed and compiled over years to run on later generations of IBM i.
  IBM i also architecturally separates the address spaces of user applications and the operating system, so it is practically impossible for poorly written, or malicious, software to intrude upon the underlying operating system support. This is one of several attributes

which keep IBM i running for days, weeks, months without unexpected restarting of the operating system.

- **Security**
  IBM i and its predecessors have an object-based architecture. Each entity on the system is an object, which has a set of prescribed operations that can be performed on it. Connections from object to object are allowed to perform various operations. An object that does not have a legal “edit” operation cannot be edited; it cannot be renamed to another kind of object. If the object does have an “edit” operation (known as “change” for most objects), it only has a defined set of attributes that can be edited only by users who have authority to do that editing.
  Objects can be easily “hidden” from other users by use of the library list objects and secured by the object based security associated with all objects. And users who all have a similar role on the system can be grouped together for role-based security, again as part of the base architecture of the system.

In conclusion

While the IBM i operating system supports open programming methods such as Java™, PHP, web services and so on, the underlying architecture provides a level of integration, stability, simplicity and security that are a significant differentiator. These aspects provide excellent business value to its users.