GUN Gotham Users of NeXT BULLETin

The New York NeXT Users Group February 1992

NeXTstep 2.1J
The Japanese Operating System

by Tim Reed

During the past year I managed application development for the Tokyo office of a large international company. Throughout that time I had the opportunity to learn some Japanese, and to use Canon's version of *NeXTstep*, *NeXTstep* 2.1J. At that time, we ran 2.1J on a 68030 cube with 8 MB of memory and found the operating system software so slow that it was unusable. I recently had the opportunity to review a production copy of 2.1J on a faster 68040 NeXTcube with a 660 MB hard disk and 12 MB of memory.

Canon recently introduced *NeXTstep* 2.1J (somewhat erroneously referred to as "the Kanji operating system" since Kanji is just one part of written Japanese) in Japan; it is a faithful port of 2.1 – bugs and all – extended to include a Japanese user interface and programmer tools. Substantial modification of operating system software is necessary to support Chinese-derived languages like Japanese since they include thousands of characters. Unfortunately most computers targeted at Japan, the largest Asian computer market, suffer from significant limitation in value and convenience. *NeXTstep* 2.1J, however, is relatively easy to use and program for – even for non-Japanese-reading developers. Both NeXT and Canon offer some support for developers who wish to port existing applications to 2.1J. However developers should carefully consider options available to them in Japan and elsewhere in Asia.

Many Asian written languages are derived from Chinese. Both Korean and Japanese were borrowed from an older form of Chinese characters, or pictograms. Although many of the pictograms that make up Chinese characters recently were simplified, people who can read Japanese can also read some Chinese as well, since the Japanese pictograph system, called Kanji, was borrowed from Chinese centuries ago. Japanese includes two other written languages besides Kanji to represent sounds. The sounds can also be written with the alphabet used in English-speaking countries. These phonetic languages are unique to Japanese and do not appear in Chinese and related languages. Katakana is used mostly to represent loan words that originate outside Japan and China. The other phonetic language, Hiragana, is usually used to represent verbs, verb tense, and minor words like conjunctions and participles. Romaji is used to show Japanese pronunciations with the Western alphabet, and is often used by Japanese computer makers as the input method on standard keyboards. Hiragana can also be used to represent any word that has a Kanji equivalent. Kanji is often used to express nouns and "big ideas," and is the most difficult of the languages to learn. Katakana and Hiragana each include over 50 characters. The number of Kana is insignificant when compared to the thousands of Kanji characters, most of which have several pronunciations.

Current computers offer incompatible "standard" systems, multiple operating systems, and counter-intuitive user interfaces. DOS is the most widespread operating system software in Japan (PCs made by NEC are the most popular) but Japanese DOS – J-DOS – will not run software compiled on a standard-issue PC running MS-DOS – not even in native English mode. In fact, DOS machines made by different manufacturers are not compatible with each other; for instance, software compiled on a Fujitsu J-DOS computer will probably not run on a J-DOS machine made by NEC or IBM. Japanese PC keyboards resemble standard QWERTY keyboards, but have a very small space key (since spaces are not often used between characters in written Japanese) and several alternate keys that modify what characters or type of characters appear when keys are pressed; they often include different Kana keycaps to show what character will appear with different alternate key combinations.

Apple offers a slightly better interface. The Macintosh requires two separate operating systems – one for Western and one for Japanese software. When Mac users need to include some Japanese text in their *Illustrator* files, they must switch language environments – which actually reboots the computer with OS software supporting the language they want. If users run Western software in the Japanese environment, garbage appears in all areas of the screen where English text should be; Japanese software likewise displays interesting results on the English-language operating system where Japanese writing normally appears. Further, the Japanese Mac operating system frequently hangs and crashes. There is no fix or workaround available to Mac and PC users for any of this behavior; the software on Japanese computers just does not work very well. Japanese users have two choices: they can continue to ignore the problems and difficulties of their systems, or they can buy a NeXT.

NeXTstep 2.1J supports all the multi-language capabilities in 2.1, plus extensions and modifications for the Japanese user. All existing software runs on the Japanese system, and non-Japanese users can use a 2.1J system and not notice any difference; even third-party English fonts work (this is the fundamental and significant difference to users between *NeXTstep* 2.1J and other Japanese operating systems). Japanese users can select the language as their primary language in Preferences; Canon apparently used the rumored Preferences API to include their own Preferences module for modifying the Japanese users' environment. When in the Japanese environment, most system messages, main menus, and window title bars appear in Japanese. Users can also easily create files and folders with Japanese names, and edit files with English and Japanese in them. Four additional 16-bit fonts that include Kana and Kanji characters are included with the standard 8-bit fonts. Some of the UNIX programs were modified to support Japanese in files and filenames; *awk* and *sed* can handle two-byte characters, but *vi* cannot understand commands if the system's front end processor is enabled.

By default, characters typed on the keyboard are sent as is to an application's key window; however, hitting Command-Space turns on Kanji

conversion via the ClareServer front end processor. The ClareServer manages input between the user's keyboard and the key application; ClareServer converts Romaji as it's typed to Hiragana – the user may use the space bar to cycle through Katakana and Kanji representations of that Hiragana. The user may hit the Return key when the character on the screen is acceptable, and start typing the next character. Typing Romaji and selecting correct representations is very quick, and, according to native Japanese computer users, more natural and intuitive than other systems. The ClareServer[1] has enough smarts to guess more common Kanji first (since there are many different pronunciations, or yomi, for each Kanji, some lookup table must exist to prioritize more popular characters); when I type "Nan desu ka" in Romaji – which means "What is this?" – ClareServer converts my input to Hiragana. If I want to see Kanji representations of the Kana and hit Space, the Clare server guesses the most popular Kanji interpretation.

Developers must develop applications differently if they wish to support Japanese in the future. Canon has added several methods to the AppKit that handle strings on character-by-character rather than byte-by-byte basis, all preceded with a CN identifier to avoid namespace conflicts. Since characters in the standard operating system are one byte, developers can assume that moving one byte up will take them to the next character; this is not true with Japanese strings and streams, since Japanese characters occupy two bytes. I do not know why this great idea was not included with the original version of *NeXTstep*, but NeXT should include it in the future. Substituting Japanese strings for original strings in labels and alert panels is easy using the macros and methods available in the Localization kit (NeXT's Localization Kit is available free from GUN); examine the source code to the incredible BackSpace application for an excellent example of localization.

Developing and using on 2.1J is not without problems. Very limited English online documentation on the Canon extensions came with my review system; I do not know if this is shipped to all users. I pity any non-Japanese reader that has to set a 2.1J system up on an existing 2.1 network with such limited documentation. I understand from other users that 2.1J runs very slowly on color NeXTstations with 12 MB of memory, although the system seems very peppy in black and white for both Japanese users running the ClareServer and for American users running without. As far as I know there is no 2.1J tech support for non-Japanese users. Canon has a helpful and energetic staff in Japan that speaks some English and responds reasonably quickly to questions it receives via fax. Both *Interface Builder* and the Objective-C compiler do not support Japanese methods, class, functions, or variable names, nor may file modules include Japanese in the names. None of the /NeXTDeveloper/Demo applications support Japanese, nor is there a Japanese online dictionary. There is no easy way to switch between English and Japanese Workspace environments without using separate user accounts – a user inconvenience, administration hassle, and security nightmare (even the language selection list in Preferences doesn't offer Japanese as one of the choices). Alert panels and voice alerts brought up by the Workspace are always Japanese regardless of the language preference. Finally, we were unsuccessful in trying to halt the Japanese OS cleanly from the Workspace; it seems that some daemon or other process always hangs the system.

There is very little encouragement for developers and consultants, especially for non-Japanese, to start using and promoting the system, since there is no developers' program available from Canon. 2.1J costs about $2,000, an astounding cost for software with no English documentation. NeXT's success is a result of its close relationship with and encouragement of its developers. Computer manufacturers know that a good computer with one or two pieces of commercial software is useless; the number of developers supporting Japanese won't increase unless Canon adopts more reasonable developer prices. Further, their changes to the operating system won't be adequately end-user tested, nor will they properly evolve and mature, unless the developers whose suggestions helped mold *NeXTstep* are encouraged to do the same for Canon's 2.1J. Developers that consult to large corporations often promote what they use themselves; without a developers' program similar to NeXT's, most corporate consultants that might otherwise suggest 2.1J, won't.

The most serious missing component for developers eyeing the Japanese market are customers for third-party applications; there are none. The high price of the NeXT Computer in Japan is a good reason why – prices are a few thousand dollar above U.S. list price, with no discounting or secondary outlets available. Other than a large university sale, there are no announced large sites running the Japanese OS. Most NeXT users in Japan buy their computers from U.S. outlets at more reasonable prices and import them. Porting and sales are currently done by Canon, which saves developers from having to provide support to non-English-speaking users, but also cuts them off from potential customers for future direct sales or custom consulting.

2.1J should be viewed in the larger approach that NeXT has taken with its operating systems. 1.0 was targeted specifically towards higher education (I remember when NeXT said that they would only sell to university students). All developers I knew and their parents were running beta versions of 2.0, an operating system that was clearly designed to answer developer requirements. Now, the rumor mill says that only very large sites are running early versions of 3.0. The new OS should be seen as the operating system targeting the requirements of large installations; this must also true of the Japanese OS – developers should become familiarized with the *NeXTstep* development environment and Japanese market requirements now, in order to better sell their software in greater volume to end-users with release of 3.0J. Developers have received some help from Canon just as developers received help from NeXT – but only if that help would result in aiding Canon computer sales.

Conclusion

NeXTstep 2.1J goes beyond the expectations of the today's Japanese computer user, but falls short of meeting NeXT's own high standards. Japanese includes thousands of characters and four written languages, requiring modifications to standard English-language computer software. Personal computer makers responded to these requirements with non-standard, incompatible, and counter-intuitive operating systems. Canon and NeXT engineered 2.1J to run both Japanese and Western applications at the same time. The ClareServer sits between the keyboard and the rest of the system by converting Romaji to Japanese characters and sending them to applications that support Japanese. Support for international users is provided by the Localization Kit from NeXT, and the 2.1J AppKit includes important extensions that handle strings character-by-character instead of byte-by-byte. 2.1J is not a mature, "production" operating system since the Japanese environment and extensions are not fully integrated with the standard operating system. Fully debugging the OS may take longer to accomplish, since Canon has not instituted a developers program to encourage developers to program for it, and has not introduced low prices to encourage users to purchase it. Developers should not expect to recover their development costs until the number of Japanese end-users increases. Developers should modify their software to support *NeXTstep* 2.1J now, so that they have an understanding of the Japanese market and relationships with Japanese distributors and vendors when *NeXTstep* 3.0J goes on the market to end-users.

(*continued*)

For More Information

NeXT, Canon, and third parties offer some assistance to people interested in the Japanese operating system. Max Henry, who is in charge of NeXT's Pacific region, is extremely helpful in providing help and information to customers and developers, and can be reached at NeXT for more information. Arthur C. Kyle of Skylee Systems, a former Canon employee and now a writer, publisher, and independent consultant, can also put you in touch with the right people at Canon; his email address is ack@skylee.com. David Thompson organizes one of the NeXT user groups in Tokyo, and can probably offer you other insights and contacts; he can be reached at 81-3-3772-6129.

---

[1] I have no idea what this stands for. *Clare* was apparently a trademark Canon already owned on an unrelated "Canon Language" product – hence the term *ClareServer* was born. On a related note, the acronym FEP – for front end processor – appears throughout the Canon AppKit extensions; FEP ranks with DASD on the list of "Most Annoyingly Inexplicable Acronyms." [I'd add CP/M to that list. –Ed.]

From the *Merriam-NeXT* Dictionary...

by Eric Rosenthal

The New Hacker's Dictionary, edited by Eric Raymond and published by the MIT Press in September, includes these definitions:

BiCapitalization n. The act said to have been performed on trademarks (such as NeXT, NeWS, *VisiCalc*, *FrameMaker*, *TK!solver*, *EasyWriter*) that have been raised above the ruck of common coinage by nonstandard capitalization. Too many marketroid types think this sort of thing is really cute, even the 2,317th time they do it. Compare *studlycaps*.

cube n. 1. [short for 'cubicle'] A module in the open-plan offices used at many programming shops. "I've got the manuals in my cube." 2. A NeXT machine (which resembles a matte-black cube).

insanely great adj. [Mac community from Steve Jobs; also BSD UNIX people via Bill Joy] Something so incredibly elegant that it is imaginable only to someone possessing the most puissant of hacker natures.

Objectionable-C n. Hackish take on "Objective-C", the name of an object-oriented dialect of C in competition with the better known C++ (it is used to write native applications on the NeXT machine). Objectionable-C uses a Smalltalk-like syntax, but lacks the flexibility of Smalltalk method calls, and (like many such efforts) comes frustratingly close to attaining the Right Thing without actually doing so.

ping [from the TCP/IP acronym 'Packet INternet Groper', prob. originally contrived to match the submariners' term for a sonar pulse] 1. n. Slang term for a small network message (ICMP ECHO) sent by a computer to check for the presence and aliveness of another. Occasionally used as a phone greeting. See ACK, also ENQ. 2. vt. To verify the presence of. 3. vt. To get the attention of. From the UNIX command ping(1) that sends an ICMP ECHO packet to another host. 4. vt. To send a message to all members of a mailing list requesting an ACK (in order to verify that everybody's addresses are reachable). "We haven't heard much of anything from Geoff, but he did respond with an ACK both times I pinged jargon-friends."

The funniest use of 'ping' to date was described in January 1991 by Steve Hayman on the USENET group comp.sys.next. He was trying to isolate a faulty cable segment on a TCP/IP Ethernet hooked up to a NeXT machine, and got tired of having to run back to his console after each cabling tweak to see if the ping packets were getting through. So he used the sound-recording feature on the NeXT, then wrote a script that repeatedly invoked ping(8), listened for an echo, and played back the recording on each returned packet. Result? A program that caused the machine to repeat, over and over, "Ping... ping... ping..." as long as the network was up. He turned the volume to maximum, ferreted through the building with one ear cocked, and found a faulty tee connector in no time.

New Element Discovered At Bellcore Research Center

by Anonymous

Red Bank, NJ (AP) The heaviest element known to science was recently discovered by physicists at the Bellcore Navasink Research Center. The element, tentatively called Administratium, has no protons, 125 assistant neutrons, 75 vice neutrons, and 111 assistant vice neutrons. This gives it an atomic mass of 312. These 312 particles are held together in a nucleus by a force that involves the continuous exchange of meson-like particles called morons.

Since it has no electrons, Administratium is inert. However, it can be detected chemically as it impedes every reaction it comes into contact with. According to the discoverers, a minute amount of Administratium caused one reaction to take four days to complete, when it would normally occur in less than one second.

Administratium has a normal half life of approximately two years, at which time it does not actually decay, but instead, undergoes a reorganization in which assistant neutrons, vice neutrons, and assistant vice neutrons exchange places. Some studies have shown that the atomic weight actually increases after each reorganization.

Research at other laboratories indicates that Administratium occurs naturally in the atmosphere. It tends to concentrate at certain points, such as government agencies, large corporations, universities, Telephone Companies, and Bellcore locations, and can actually be found in the newest, best maintained buildings. WINDOWLESS LAB OFFICES APPEAR ESPECIALLY SUSCEPTIBLE TO INFESTATION.

Scientists point out that Administratium is known to be toxic at any level of concentration and can easily destroy any productive reactions where it is allowed to accumulate. Attempts are being made to determine how Administratium can be controlled to prevent irreversible damage, but results to date are not promising.

NeXT In The News

by Eric Rosenthal

"The Ghost From the Grand Banks", Arthur C. Clarke [Bantam Books, 1990, p. 55]: The noted science fiction author described a character's activities in 2007: "Presently he sat up, switched off the music module,