

Ascension

After Unicode: Fidel's Roadway to Higher Ground in Academic and Scientific Computer Environments

by Daniel Yacob

Computers in 1996 Ethiopia are no longer just the different breeds of IBM PC clones or Macintoshes. A new word is rolling off the tongues now of computer affections in Addis Ababa when they verbally joust one another on the merits and perils of Macs, Windows 3.1 or '95, or perhaps even OS/2. It is 1996 and many PC users across Ethiopia have started using an operating system called "Unix" -whether they realized it or not.

On May 21st the Hornet BBS of the ECA/PADIS was reopened on a Unix server and everyone with a PC and a modem was welcome to come explore new services, such as local World Wide Web, that were not possible previously on the DOS system. The Hornet host is one of at least a half dozen Unix systems operational around town now. This includes most significantly the School of Information Studies in Africa (SISA) at Addis Ababa University and the ESTC (Ethiopian Science and Technology Center) where pivotal new work may soon emerge. The count will likely grow after the ETA begins and later expands its own Internet service.

But why use a new computer system? Is it required for Internet? First, Unix is anything but a new operating system, in fact it predates DOS. Unix is not required for Internet but the operating system is much more capable of handling multiple users, processes, and the higher traffic flows that come along with providing an Internet service. This makes both customers and computer managers happier.

When the microcomputers became affordable in the early 1980s, they were given an operating system based on Unix but tailored to meet the micro computer's limited capacities. This operating system was DOS. Fifteen years later these micro computers (PCs) are not as limited as they once were (RAM memory for example has typically grown from 48K to 8 Meg) and are now more capable of supporting a fully featured Unix operating system -once reserved solely for the main frame super computers and work stations.

Traditionally the computers that would operate on Unix systems would be reserved for the research purposes of scientists and engineers. While computers operating on DOS both created and filled the market niche of the small business and the home user. The DOS market proliferated and gave rise to new applications like the spread sheet and higher quality word processors that the scientists and engineers would later find useful.

The commercial arena of the personal computer has been the fertile grounds where Ethiopic script has been able to grow and prosper in electronic media. Despite the limitations and difficulties of working with Fidel on DOS and MS Windows systems; ingenuitive and determined Ethiopian computer experts persevered and found ways around would be barriers.

Until recently, the noncommercial arena where most scientific research goes on was left untreaded by the Ethiopic script. This is understandable but very unfortunate none the less. The Unix operating system was always very hospitable to Fidel -never presenting the 128 or 256 letter limits of DOS & Windows. For scripts that did force their way into the Unix world; Japanese, Chinese, Korean, French, Arabic, Russian, Hebrew, etc., now have a very comfortable home on the latest communication medium -the Internet.

Because these scripts were available to researchers in the government and in academia, more could be done to study the issues important to communication with the scripts and the languages that use them. The research that occurred in university and research center labs would later find its way back into the market place.

It is not important nor of any value to look further into why Ethiopic was left behind in the Unix arena when other scripts advanced and flourished. What serves Ethiopic -the writing systems, the languages, communication, and those who wish to communicate; is to look at what is required to keep it on these systems and be carried along with each wave of advancement. To then fill the requirements is a duty of the willing and the able.

At this point we need to take a step back from the operating system (Unix or VMS) and focus at the more macroscopic level of the work environment. The work environment may be described not only the windows system we are using and our personal attributes preferences; but also, by the shells we work in, the applications we work with (editors, mail tools, web browsers, news readers, compilers, spell checkers, etc.), the way we communicate with the environment (typing), and the way it communicates with us (menus and messages).

When a multi-lingual environment or application is set to work with the expectations of a specific language; the resource is said to have become “localized”^{*} for that language. To date, an Ethiopic work environment has yet to be defined -which stems from the lack of readily available resources to define one. Our work starts here.

Resources

The smallest element of human-to-computer and computer-to-human communication is the same element that allows you and I to communicate now; the character. Purchasing a new computer the characters (letters) come with the computer. They must, or the computer is rendered even more unusable than a computer without a keyboard. In the workstation and mainframe arena the defacto graphical environment (think “Windows 2015”) is based on the X11 Windows protocol. The distinction between a windows environment and protocol is not important now. X11 is a freely available windows system that originates from the laboratories of MIT and is now maintained by the X Organization.

With X11 comes all of the characters one would need for communication in Japanese, Chinese, Hebrew, Korean, Arabic, etc. The professors at MIT did not spend their valuable time to hand edit 6 varieties of the more than 5,000 Korean characters. They didn't have to. Realizing the fundamental importance of a character set to the valuable work of people speaking a given language; companies such as Adobe, Sony, Sun, B&H, etc. donated character sets (naturally in the form of fonts) to the caretakers of X11.

^{*} The term “Localization” is popularly written with the short-hand “L10N” for “L” + 10 letters + “N”. Similarly “Internationalization” becomes “I18N”. It would be in keeping with the L10N and J10N (“Japanization”) convention to apply the prefixi “Ethio-”, “Fidel-”, or “Ge’ez-” to from “E10N”, “F10N”, and “G10N” accordingly. The author prefers “G10N” as “F10N” implies L10N of the writing system only. “Ethio-” is appropriate as it is applied very broadly, even beyond the languages and the writing system. “Ge’ez-” may lack word recognition in the western world but it is less encompassing and implies a root, an origin from which most things considered Ethiopic descend. G10N should be considered to mean “localization for those languages whos' writing systems descend from Ge’ez script”. In general, the three choices could be applied interchangeably without invoking confusion.

With X11 on nearly every workstation and mainframe with a window system, so too would be these character sets. The availability of which allowed communication on Internet to be possible and for applications that could use them to be developed. In 1996 an initiative began to do the same for Ethiopic. Goha Tibeb, EthiO Systems, Admas Concepts, and the ESTC/NCIC have contributed fonts to become a part of the next X11 release.

With the assured availability of Ethiopic fonts and an internationally recognized character coding standard (i.e. Unicode) developers may proceed with confidence to start on new work. Indeed, some of these public fonts have already found their way into the big engine applications of the X11 environment such as; LaTeX, Netscape, Emacs and experimentally in IRC. Distribution with the Java language interpreters may also become a reality.

At these modest levels the applications shoulder the burden of keyboard entry interpretation. To take advantage of all of the software running in the environment, even software not designed with fore knowledge of Ethiopic, the keyboard input method (IM) must become application independent. Under X11 a protocol system is provided to remove the task of IM from the application. Implementation of an XIM may be through an IM server, running as a separate process locally or over a network; or as shared library called up by the application at run time.

Though the facility is provided for Ethiopic IM, no serious work has been under taken to apply existing IMs (developed on PCs) under X11 protocols. Thus, even with fonts now available, applications that are more than file viewers remain unavailable until this work initiates.

The fonts and IM are fundamental necessities but still only a part of a much larger work in wait for the community to pursue. A complete Ethiopic Languages User Interface (ELUX) would be the natural goal of the G10N efforts. An ELUX specification would describe not only font sets and IM but also the appropriate punctuation to format time and currency, text I/O, the Ethio-Julian calendar system, and application, messages, command vocabulary to use for different language selections, and perhaps even default colors, pointers, icon, and bullet glyph preferences. An example for the vocabulary specification would be the preferred Amharic or Orominga expressions for “Save As...”, “Paste”, “cd”, and “No such file or directory”, etc. ELUX may be implemented as part of an IM server or library, and in the X11 and operating system's locale databases. An ELUX specification also assures consistency of the interface across applications and platforms.

It must be realized that much of the work that may later be done to localize Ethiopic languages in software may be carried out by developers at companies and at organization who have no functional knowledge of any of the languages or have any ability to read the script. To assure that as many internationalized applications are given L10N for Ethiopic as can be achieved, the work and learning cycles of this class of developers must be brought to a minimum.

The learning cycles may be reduced by providing public information recourses on the Internet. Examples of this being done are already present. Minimizing the efforts of all developers, the informed and uninformed, would be a computer library in the public domain that offered the routines for performing G10N tasks. A library in ANSI C for just this is “LibEth”, that 4 Unix applications already depend on. LibEth is young yet promising in principle. Like the public domain fonts, programmers are freed from writing the mundane and tedious routines that they may not be willing to compose, and may proceed on with tackling their methods' logic. As this article goes to press, LibEth is a useful but incomplete resource, its completion should be addressed prior to the composition of computer codes for XIM and ELUX efforts. LibEth's usefulness for rapid application development and for adding G10N in existing software need not be limited to the work station and mainframe laboratories. Ports of the library to DOS and other operating

systems, and to other computer languages, would be an immeasurable benefit to the Ethiopics software community.

There are no end to the number of research and commercial applications that could be created for or adapted to the languages using the Ethiopic writing system. Providing the computational resources in the public domain is the best way to assure G10N happens in as many instances as possible. Building the resources is then the greatest service that could be performed for Ethiopics in computers in 1996 and for every year afterward.

Epilogue

The projects described in this article are in their infancy and are in need of skilled programmers who can volunteer their time and skills to help the realization of these efforts. Interested parties, and groups or individuals working in like areas, may contact the author for coordination of efforts.