

## 0x00: Introduction

On a hot summer day, while the rest of the industry was overloaded fixing Y2K bugs, Cloanto's Software & Dreams department took a day off to design a series of T-shirts.

Cloanto T-shirts were originally meant "for internal use only". However, it immediately became clear that people would do anything in order to get one. Very soon, Cloanto's software engineers could be seen in exotic beaches and elegant downtown squares around the world, offering T-shirts in exchange for antique computers, eternal friendship, sex and other mundane distractions...

Cloanto's Work, Work & Work department quickly became aware of the problem, and decided to look at new ways to make these T-shirts available to a larger public without diverting precious engineering resources. The results can be seen (and purchased) in our T-shirt stores (North America, Europe). All designs are produced using high-quality and extremely durable multi-layer plot printing. In comparison, digital printing is not as smooth and brilliant, and tends to fade and crack. Plot printing is also more durable than offset printing.

Each T-shirt design features a different message written by human programmers and output by machines. The following sections explain the story behind each message. In true programmer fashion, the different T-shirt styles are numbered using hexadecimal notation ("0x...").



## 0x01: /\* You are not expected to understand this \*/

In the C programming language `/* ... */` denotes a comment in the software source code. The "You are not expected to understand this" comment could be found in the C source of the Sixth Edition Unix operating system, released in 1975.

Bell Laboratories' Dennis Ritchie, credited with creating both C and Unix, and now a proud owner of this T-shirt, explains:

It's often quoted as a slur on the quantity or quality of the comments in the Bell Labs research releases of Unix. Not an unfair observation in general, I fear, but in this case unjustified. The actual code and other commentary surrounding it were precisely this:

```

/*
 * Switch to stack of the new process and set up
 * his segmentation registers.
 */
retu(rp->p_addr);
sureg();
/*
 * If the new process paused because it was
 * swapped out, set the stack level to the last call
 * to savu(u_ssav). This means that the return
 * which is executed immediately after the call to aretu
 * actually returns from the last routine which did
 * the savu.
 *
 * You are not expected to understand this.
 */
if(rp->p_flag&SSWAP) {
    rp->p_flag =& ~SSWAP;
    aretu(u.u_ssav);
}
/*
 * The value returned here has many subtle implications.
 * See the newproc comments.
 */
return(1);

```

So we tried to explain what was going on. "You are not expected to understand this" was intended as a remark in the spirit of "This won't be on the exam," rather than as an impudent challenge.

The real problem is that we didn't understand what was going on either. The savu/retu mechanism for doing process exchange was fundamentally broken because it depended on switching to a previous stack frame and executing function return code in a different procedure from the one that saved the earlier state. This worked on the PDP-11 because its compiler always used the same context-save mechanism; with the Interdata compiler, the procedure return code differed depending on which registers were saved.

So, for Steve Johnson and me, trying to move the kernel for the first time to a new machine, this code was indeed on the exam. It took about a week of agonizing before we finally convinced each other that the mechanism was wrong and no fiddling with the compiler was useful. We redid the coroutine control-passing primitives altogether, and this code section, and the comment, passed into history.



## 0x02: values of b will give rise to dom!

One of the original major applications of the Unix operating system was the processing of patent texts. During one of these tasks this mysterious message first appeared on a Teletype printer. In the Sixth Edition of Unix this message was incorporated into the "mv" command, which is used to move or rename files.

If your browser cannot display the Greek letter in the quote, please note that the text should read "values of beta will give rise to dom!"

Dennis Ritchie explains:

The actual source line in mv.c that produced the message was

```
write(1,"values of \016B\017 will give rise to dom!\n",37);
```

except that in the real source, the \016B\017 was written with literal ASCII SI and SO control characters. These shifted the Model 37 Teletype into its optional extended character set, and the B printed as the Greek letter beta.

Like most of the messages recorded in these compilations, this one was produced in some situation that we considered unlikely or as result of abuse; the details don't matter. I'm recording why the phrase was selected.

The very first use of Unix in the "real business" of Bell Labs was to type and produce patent applications, and for a while in the early 1970s we had three typists busily typing away in the grotty lab on the sixth floor. One day someone came in and observed on the paper sticking out of one of the Teletypes, displayed in magnificent isolation, this ominous phrase:

```
values of b will give rise to dom!
```

It was of course obvious that the typist had interrupted a printout (generating the "!" from the ed editor) and moved up the paper, and that the context must have been something like "varying values of beta will give rise to domain wall movement" or some other fragment of a physically plausible patent application.

But the phrase itself was just so striking! Utterly meaningless, but it looks like what... a warning? What is "dom?"

At the same time, we were experimenting with text-to-voice software by Doug McIlroy and others, and of course the phrase was tried out with it. For whatever reason, its rendition of "give rise to dom!" accented the last word in a way that emphasized the phonetic similarity between "doom" and the first syllable of "dominance." It pronounced "beta" in the British style, "beeta." The entire occurrence became a small, shared treasure.

The phrase had to be recorded somewhere, and it was, in the v6 source. Most likely it was Bob Morris who did the deed, but it could just as easily have been Ken.





### 0x03: Keyboard not present. Press any key.

This error message is an example of what can happen when a standard text (in this case "Press any key [to continue]") is automatically appended after an error message, without consideration to what the first part might actually be. In this case the BIOS code of a PC, as part of some error checking performed after the computer is switched on, detected that the keyboard was missing, and displayed "Keyboard not present. Press any key."

This and other similar messages inspired a variety of computer tales and jokes targeting programmers (the message is an obvious programming mishap), computers ("PCs are stupid") and users (for example, a computer company reportedly received support calls asking where the "Any" key is).



### 0x04: You can't do that in horizontal mode.

"Horizontal Mode" is one of six different modes that affect processing and rendering in the TeX typesetting system, and is used for constructing paragraphs of text. In this context, "You can't do that in horizontal mode" is a self-explaining error message.

The TeX software was designed by Donald Knuth, a professor of computer science at Stanford University and the author of numerous books on computer science. "TeX" is a creative transliteration of the Greek letters tau epsilon and chi, from a Greek word root that can mean both art and technology, and is pronounced *teck*.

## 0xAA: Codex Alpe Adria 2006

To celebrate Codex Alpe Adria 2006, the event which we help organize, we created a special T-shirt.

### /\* North American vs. European Store \*/

The T-shirts are available from two different stores: North America and Europe. The stores are run by two sister companies of the Spreadshirt group which carry a slightly different product range. The US warehouse offers a better selection of T-shirt sizes for all our designs, while the European store has a more limited choice of larger sizes for the orange T-shirts. On the other hand, the European store allows us to make the T-shirts available both in the original version and in a high-tech variant where a special reflective material ("power reflex") is used for the white parts.



In order to save on shipping costs we would normally recommend to select the store nearest to you. However, if you prefer the more high-tech (and slightly more expensive) reflex print, or if you need a larger T-shirt size, you may base your choice on that. Please note that European prices are in euros and include VAT, if applicable to your country.

*The Legal department, jealous of the success of the Software & Dreams and Work, Work & Work departments, would like to add: "All designs are Copyright © 1999-2006 Cloanto. Cloanto is either a registered trademark or trademark of Cloanto Corporation in the United States and/or other countries. All information is subject to change without notice." And our Quality Assurance department would like to add that we use only the best garment and printing technologies. Do you already own a Cloanto T-shirt? Why not send us a link to a picture of yourself wearing it!*