Programmers get it in the neck from Brian Clegg for not taking into account human error.

Only human

If the computer fairy were to grant me a single wish to make life better for the users of business PCs, it would be to make a small but significant modification to programmers' brains. I'd like them to be given the innate ability to make allowances for human error, the kind of human error that is entirely predictable. The improvement would be remarkable.

Take the plight of a poor computer user I was speaking to the other day. Writing for a magazine means you often get asked which PC to buy ('check out the latest *PCW*') or told about the way a computer has misbehaved. This long-suffering user had a piece of bespoke software that could operate in two modes. If you tried to enter data in the wrong mode, it said words to the effect of 'you are in reporting mode; to do this you should be in entry mode' (though without the punctuation). The user then had to perform about ten key-presses to change modes. Given that the system knew it was in the wrong mode, it should, my new friend suggested, have switched automatically to the right one. And why not, as the irreplaceable Barry Norman might have said before his disappearance into the obscurity of satellite broadcasting.

I challenge any Unix fan to specify the benefit to human beings of MAKING FILENAMES CASE SENSITIVE. It is nothing short of stupid

The perpetrator of that example was certainly thoughtless, but I've a different candidate to be the first against the wall when the revolution comes — whoever decided that Unix should be case sensitive, thinking that a word beginning with 'a' was totally different from the same set of letters beginning with 'A'.

Until recently this has only been a problem for those who have chosen to use Unix, the sort of people who go for the macho option to prove what real men, women or strange grey entities they happen to be. But the ubiquity of Unix on the world wide web has made it a problem for everyone. Hands up all those who have put up a tasty new image on their website, only to have nothing appear? Why? Because the HTML code required 'BRIAN.JPG' but the file was actually called 'brian.jpg'.

If experience is anything to go by, I'm treading on dangerous ground when criticising an operating system. I have had hate mail in the past for not liking OS/2 — need I say more? Yet I challenge any Unix fan to specify the benefit to human beings of making filenames case sensitive. It is nothing short of stupid. I can see that it makes life easier for the computer: as far as it is concerned, there is no obvious connection between 'a' and 'A'. Yet the operating system was developed by and for English-speaking human beings, and to distinguish words by the case employed is entirely inappropriate.

'Ah,' says the security-conscious reader (or possibly 'ah', 'aH' or 'AH'), 'what about passwords, though? We ought to use a mix of case in our passwords.' Sorry, it still won't do. Case-sensitive passwords are an abomination that the European Parliament should scrutinise before worrying about web cacheing and defamatory postings in newsgroups. The amount of support effort that goes into chasing up login problems that are caused by having Caps Lock on is ludicrous.

How can you possibly expect someone to use mixed case accurately when they can't even see what they are typing? What's more, I'd argue that the use of case-sensitive passwords decreases security. The conventional argument says that having case sensitivity makes the password harder to guess. Unfortunately, it makes it impossible to remember, as memory does not normally deal with words in terms of case. Result? The password gets written down and security is compromised.

If programmers could concentrate on just those two issues, the outcome would be more valuable than a whole new spate of features and functions. Eliminate case sensitivity, unless human beings are never involved. Each time you produce an error message, double-check the meaning. If it's 'you can't do that here', don't just tell them, take them to where they *can* do it. If you know what they are trying to do, make it happen. Remember you are in the service business. If you can't help, someone else will.

Deliberately making software unhelpful may provide some perverse pleasure for a programmer with a dislike of the less technically gifted, but it also means that he or she is not doing a good job. Error messages should be there to help, not insult.

BrianClegg@cul.co.uk