Chapter 8

Naive-user documentation

8.1 Goals—better paragraphs and writing for non-technical audiences

This assignment has two purposes—to improve your paragraph structure and to make you appreciate how difficult writing for an ignorant audience is.

Huckin and Olsen [HO91] have an excellent chapter on paragraphing, Chapter 22, and another on parallelism, Chapter 23. Section 12.1 has good general ideas about engaging more-or-less ignorant audiences. This is also a good time to read Chapter 3 (Identifying Audiences and Purposes). Of course, Chapter 17 (Instructions, Procedures, and Computer Documentation) is worth reading for this assignment.

Notice how the paragraph you have just read is parallel in structure to the introductory one that precedes it. This is an example of an important kind of parallelism, and is discussed in Huckin and Olsen [HO91, Section 23.3].

Also notice that Section 8.1 of this workbook has rather short paragraphs. Short paragraphs are one of six important formatting conventions for making reading easier [HO91, 177–179]. Read Sections 9.3 and 9.4 and Chapter 21 of Huckin and Olsen for more suggestions on increasing readability [HO91].

Finally, take a good look at Scott Brookie’s *Unix for Luddites* [Bro86]. This manual is a classic for introducing a complex system to an audience who is not only ignorant, but hostile. Brookie recognizes, and attempts to defuse, this hostility—his first sentence is:

Luddites are people who believe that new kinds of technology will make their lives worse.

Huckin and Olsen discuss introducing a problem with two directly conflicting terms in their Chapter 5. Brookie’s sentence is a rather subtle version of this. After you have read Huckin and Olsen’s Chapter 5, see if you can figure out how Brookie’s first sentence, and the two introductory paragraphs of his manual, work to make effective contact with the naive user.

8.2 Audience assessment—non-technical audiences

You have been hired as a training consultant by the Boring Business Company. Boring Business has decided they need to build and maintain a web-site, but they don’t want to hire a lot of designers. One employee suggested that everyone in the company have a web page, to give the company a more human face.

They have mainly UNIX workstations for their employees, and so most of the commercial software for beginner’s to create web sites is not really useful to them.

You have to write a training manual explaining how to use HTML to create a personal web page.

Most of the readers will be typists and receptionists, but some executives at Boring Business will glance through the manual so that they can pretend they created their own web pages, rather than having a secretary do it for them. You can assume that all your readers know how to use a keyboard, but at least some of the readers will not have looked at any other training manuals.

You do not need to explain all the details of HTML and creating web pages, only as much as a person needs to create a simple personal web page. You can use any existing documentation as a reference, but be sure to cite it properly. There are many HTML references on the web itself—it is a good (though easy) exercise to use web search
8.3 Writing process—paragraph structure

Paragraph structure is an aspect of writing that used to be taught in grade school and high school, but seems to have disappeared from the American school curriculum. Students who learned English as a second language may never have gotten past sentence structure. Fortunately, proper paragraph structure is not complicated and can be learned quickly.

For programmers, there is a useful analogy between program structure and document structure. A sentence is analogous to a program statement, and a paragraph to a procedure. Like a procedure, a paragraph should have a single well-defined purpose. Just as a procedure begins with a block comment explaining its function, a paragraph begins with a topic sentence that gives the main subject of the paragraph.

Although some writing texts mention other possible paragraph structures, for technical writing the topic sentence should always be the first sentence. Check the first sentence of each paragraph to see if it says what the paragraph is about. If it doesn’t, try to decide whether the sentence should be fixed to match the paragraph, or the paragraph fixed to match the sentence.

The two most common paragraphing faults are rambling paragraphs and artificial breaks. A rambling paragraph has several topics, often not closely related. It can be fixed either by breaking the paragraph into smaller ones or by removing extraneous material. Artificial breaks occur when novice writers try to divide paragraphs that are too long. The lack of topic sentences defeats the entire purpose of having paragraphs.

The proper length for a paragraph varies, depending on the content and the importance of the information. Three to five sentences (about 30 to 60 words) is a common length. The topic sentence is supported by a few, well-chosen, additional sentences. If your paragraph runs on and on, your readers may glance at the first sentence or two, then skip to the next paragraph. Never bury the key points in the middle or end of a long paragraph.

Short paragraphs make emphatic statements. Use them sparingly.

The continuity between paragraphs is as important as the internal structure of the paragraphs. Sudden changes of subject will make readers stumble, but choosing a suitable ordering for your topics will often let you flow smoothly from one to the next. When no ordering seems to work, try breaking the text into separate sections, each on a separate subject.

Within a section, paragraph breaks can be made less abrupt by anticipating the next subject at the end of one paragraph, or by referring to the previous subject in the topic sentence of the new paragraph. Notice how abrupt this paragraph would have been if I’d left off “Within a section”.

Paragraphing is especially important in naive-user documentation, because documentation is consulted, not read like a novel. Related information must be brought together into one paragraph or into contiguous paragraphs, so that users looking for one piece of information will see the essential related material. Using headlines for sections (as in this workbook) helps guide the user to the right place.

8.4 Things to keep in mind for editing and partner work

Students usually write for a single audience, their teacher. Because the teacher usually knows more than the students, the purpose of the writing is usually to impress, and not to inform. The teacher’s purpose is to find out whether you learned anything, and, sometimes, whether you can articulate your thoughts clearly. Rarely is the teacher interested in the ostensible content of the paper.

In this assignment, you are writing for a different audience, one presumed to know less than you. Your readers will not know all the jargon you use almost unconsciously, so you’ll have to be extra careful about which words you use. You’ll also have to be careful about the order in which you present new ideas. New computer users do not want to have to read documentation twice—most don’t even want to read it once.

Not everyone will choose the same features of HTML to cover in an introductory document. Consider carefully the intended purposes of the web pages before deciding whether to describe particular features. You should make your decisions based on how important it is for the users to be able to use the features, not on how easily you can describe them.
<table>
<thead>
<tr>
<th>word</th>
<th>computer usage</th>
<th>normal usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>software instruction</td>
<td>cryptic message</td>
</tr>
<tr>
<td>boot</td>
<td>load operating system</td>
<td>footwear</td>
</tr>
<tr>
<td>virus</td>
<td>makes computer sick</td>
<td>makes you sick</td>
</tr>
<tr>
<td>memory</td>
<td>data storage</td>
<td>retained ideas</td>
</tr>
<tr>
<td>news</td>
<td>Usenet</td>
<td>NBC/CNN/C-Span</td>
</tr>
<tr>
<td>mail</td>
<td>electronic letters</td>
<td>bills/junk mail</td>
</tr>
<tr>
<td>FIDO</td>
<td>subnet</td>
<td>dog</td>
</tr>
<tr>
<td>pen</td>
<td>pointing device</td>
<td>writing with ink thing</td>
</tr>
<tr>
<td>SLIP</td>
<td>serial line interface protocol</td>
<td>a fall/undergarment</td>
</tr>
<tr>
<td>TIP</td>
<td>open line for comm.</td>
<td>money for waiters/waitresses</td>
</tr>
<tr>
<td>mouse</td>
<td>pointing device</td>
<td>rodent</td>
</tr>
<tr>
<td>screen</td>
<td>terminal face</td>
<td>metal mesh</td>
</tr>
<tr>
<td>spool</td>
<td>place to save mail or output</td>
<td>thing that holds thread</td>
</tr>
<tr>
<td>thread</td>
<td>code structure method</td>
<td>stuff on spools</td>
</tr>
<tr>
<td>OOPS</td>
<td>C++ or Ada</td>
<td>a booboo</td>
</tr>
<tr>
<td>ports</td>
<td>serial, parallel, ...</td>
<td>place where ships dock</td>
</tr>
<tr>
<td>floppy</td>
<td>removable disk</td>
<td>limp</td>
</tr>
<tr>
<td>hard drive</td>
<td>fixed disk</td>
<td>difficult trip</td>
</tr>
<tr>
<td>Windows</td>
<td>GUI nightmare</td>
<td>cleaning nightmare</td>
</tr>
<tr>
<td>root</td>
<td>UNIX system administrator</td>
<td>bottom part of plant</td>
</tr>
<tr>
<td>Smalltalk</td>
<td>programming language</td>
<td>chit chat</td>
</tr>
</tbody>
</table>

Table 8.1: Some common jargon, modified from a list posted to the newsgroup rec.humor.funny.

A one-page quick summary can be a useful adjunct to the more detailed tutorial you have written. It should be usable as a reference both by those who have read the entire chapter, and by those who think they know what they’re doing, and don’t have time to wade through the introductory stuff.

Be careful not to assume too much knowledge on the part of the user. (Will a novice UNIX user know how to use more?)

Watch out for sentence fragments! Fragments? Noun phrases without verbs. Ex-President Bush, famous for fragments. Very hard, figuring out what fragments are supposed to be saying, especially when they are as long as this one and contain verbs in subsidiary clauses.

Avoid Latin abbreviations! The most often abused Latin abbreviation is etc., and abbreviation for et cetera which means and other things. Do not say “and etc.”, and do not use etc. for people. If you must use a Latin abbreviation for and other people, use et al. (for et alia), but the use of Latin in formal writing is a relic of the era when all literate people were expected to know Latin, and all writing was done with pens. With word processors, there is little reason not to use the equivalent English expressions.

Other Latin abbreviations to avoid are i.e. and e.g. If you must use them, punctuate them properly (i.e., like this). I prefer to use the simple English phrases (for example, I use in other words for i.e.).

Watch the jargon! Using login as a synonym for log in name is a UCSC idiom that will not be understood in other contexts. In standard usage, log in as a noun refers to the process of logging in, not the person who does it. The verb is log in—like many verb-particle pairs, the verb is two words, but the corresponding noun is one. Table 8.1 gives a table of other jargon words adapted from a list posted to the newsgroup rec.humor.funny.

Another sloppy usage that seems to be common at UCSC (and perhaps more generally) is the use of substitute y with x when substitute x for y is intended.

Watch out for words beginning with any. In my dictionary, only anybody, anyhow, anymore, anyone, anywhere, anything, anyway, and anywhere begin with any. Any time you find yourself using a different word beginning with any, break it up. Sometimes, words beginning with some are spurious compounds also—a good spelling checker can be a big help.

Although English allows almost any noun to be used informally as a verb (a process known, appropriately, as verbing), formal writing does not allow this process. For example, access is a noun not a verb, you should get access or obtain access—but this abuse is widespread enough that within a decade or two it will probably become acceptable.
Another abused noun is reference, which is formed from the perfectly good verb refer.

It is perfectly acceptable to use we in a formal document, if you mean you and I or we, the multiple authors. The singular pronoun I is much more problematic, and should only be used when the human being peeks out from behind the formal facade: the first time I used UNIX mail, I managed to send blank messages to all the faculty in the department, because I didn’t understand the difference between reply with R and reply with r.

Watch out for split infinitives. For some reason, this assignment always brings out a rash of them (for example, to simply send and to just send). Many times, the offending word could be dropped from the middle of the infinitive without loss.

Some amusing dangling modifiers have occurred as well: After typing in the subject, the cursor will . . . .

8.5 The final draft

As always, the final draft should incorporate any improvements that resulted from your partner’s editing. You will probably have to add more material at the beginning of the chapter and change the order in which you cover the topics. If you use a word-processor to rearrange your first draft, be careful to redo the transitions so the seams don’t show.

The final draft should be on 8 1/2” × 11” paper, stapled together or bound on the left edge. Paper clips and slip-on plastic binders are not suitable, as pages tend to slip out of them and scatter. We generally prefer papers with one staple in the upper left corner—they are easy to handle and to photocopy.

Make sure you remove all spelling and punctuation errors. Good typists will not trust documentation that contains errors in their area of expertise.

Check your use of articles and subject/verb agreement. For example, mail is not a countable noun, and so it can not be used with plural verbs or indefinite articles. If you need to count items of mail, you can use the word messages.

Use a standard form for citing your sources of information. There is no widely accepted form for citing on-line documentation, but the UNIX man pages have been printed as a book, which can be cited.

Check that all your examples are 100% accurate. Try them out to be sure. It is very frustrating to read a manual that says “you will now see . . . on your screen,” and to have something else entirely appear. Make your examples explicit, rather than using the Backus-Naur Form notation that computer scientists often adopt for describing syntax in a generic way. People are much better at reasoning by analogy than computers, and much poorer at substituting values for variables. Give them examples they can modify, rather than rules for which they have to fill in the blanks.

When you give examples of what people are to type, or what they should see on the screen, use displays. That is, put the material on a line by itself, even though it is part of the surrounding sentence or paragraph. Punctuate the sentence normally—it is incorrect to add a colon before a display, unless the display is a list that would require a colon before it anyway. For example, this sentence has

one line displayed,

but is otherwise normally punctuated.

Since manual chapters don’t usually have acknowledgements, prepare a separate short memo to the instructors saying who helped you with the editing. Turn in both the first and final drafts. This memo should also explain why you chose to include or omit certain material from the manual.