LISTSERV for the non-technical user

September 18th, 1993

Eric Thomas
ERIC@SEARN.SUNET.SE

Royal Institute of Technology
Dr. Kristinas väg 37B
100 44 Stockholm
Sweden

Copyright Eric Thomas, 1993 - All rights reserved worldwide

Permission is granted to copy this document, at no charge and in its entirety, provided that the copies are not used for commercial advantage, that the source is cited and that the present copyright notice is included in all copies, so that the recipients of such copies are equally bound to abide by the present conditions. Prior written permission is required for any commercial use of this document, in whole or in part, and for any partial reproduction of the contents of this document exceeding 50 lines of up to 80 characters, or equivalent. The title page, table of contents and index, if any, are not considered to be part of the document for the purposes of this copyright notice, and can be freely removed if present.

The purpose of this copyright is to protect your right to make free copies of this paper for your friends and colleagues, to prevent publishers from using it for commercial advantage, and to prevent ill-meaning people from altering the meaning of the document by changing or removing a few paragraphs.

Jump start

Oh no, not *another* hernia-inducing handout! If you want to get started without having to read everything,

- Skip directly to the "What is..." chapter and read the overview of mailing lists, then the section about "LISTSERV", and possibly the description of "Filelists".
- Then skip to the "How to..." chapter and read the parts you are interested in. Everything is indexed in the table of contents for easier access. The headings were designed to give you a flash overview of the parts you won't have time to read.
- When you reach "The global view", skip directly to "The Global List Exchange (GLX)". Once you get to "Future plans for LISTSERV", it is enough to read the headings.

Foreword

When the first version of "Revised LISTSERV" (as it was then called) was released in 1986, "the net" was a much smaller world than it is today, and the fastest lines we had could transfer as much as one kilobyte of data per second, under good conditions. The US and Europe were connected to each other by two such lines, of which at least one was down at any given time, and you often had to wait a day or two for your files to cross the Atlantic. In just 7 years, the capacity of trunk lines has multiplied by 3500, and users now have access to a whole range of new applications which would have been unthinkable just 3 years ago.

But you probably knew that already, because this is a typical thing to say in the introduction of a paper about networking. What people do not usually say, however, is that the network was also a lot simpler in 1986. Oh, there were complicated system commands to learn about and obscure error messages of course, and the computer experts had already started conspiring against mere mortals. In fact, the buzzword *à la mode* was "user-friendliness",

adding insult to the injury. Mail programs that would occasionally produce error messages like *"The entire file must be peeked to be reformatted"*, with no apparent reason, and then spurt out a screenful of random, indecipherable characters were user-friendly, and so were computers that expected you to type double quotes all the time, on penalty of having your message translated to upper case. But, on the other hand, there were a lot less things to know about than nowadays. Once you knew how to use your computer and how "e-mail addresses" work, you were all set and could go back to whatever it is you normally do when not wrestling with your computer. One of your colleagues used to have a paper in his drawer with a list of "file servers" from which you could get a number of useful programs, or information about the network. You'd open the drawer a couple times a month and be reminded that well-educated people only peek at things in their entirety, and the rest of the time you would just stick to e-mail and a few other simple things that anyone can learn in a matter of days.

Nowadays, on the other hand, the buzzwords are "multimedia", "object-oriented", "virtual reality" and "Cyberspace", and any self-respecting network citizen is expected to know about a couple dozen different "services", most of which require special "clients" to be accessed - clients which, unfortunately, anyone can get via "anonymous FTP", meaning *you* are often expected to take care of that yourself. Which is not to say that things were better in the old days, or that we should just have upgraded the lines and stayed clear of new services. People are getting a lot more work done over the network nowadays than in 1986, thanks to the excellent work of the computer experts and to new services such as *gopher,* which does require a client but is well worth the effort. One does, however, occasionally wonder whether the "techies" are not going too fast sometimes, in their (understandable) desire to improve things. While it is generally agreed that future expansion of the Internet will come mostly from the corporate world, and that most of these new users will know little or nothing about computer technology, a disquieting number of "techies" are advocating changes which would make it much harder for these users to get their jobs done.

For instance, many people are lobbying for the systematic use of various MIME[(1)] features with LISTSERV, in spite of the fact that most users do not have MIME-compliant mail programs and could be seriously inconvenienced by this change (some of the most immediately useful MIME functions render the messages totally illegible to anyone not equipped with a MIME decoder). The rationale, of course, is that this will force all these people to "leave the stone age" and install a "decent" mail program, notwithstanding the fact that they might be fully satisfied with the program they have, and have no interest whatsoever in mailing recorded sound over the network. Similarly, users of so-called "office" mail packages are usually unable to issue commands to mail-based servers, because the "inter-office gateway" inserts a letterhead-like header on top of any message sent to the outside world and there is no way to tell the server to skip that letterhead, or otherwise keep looking for the user's command. The general consensus is that these programs are "brain damaged" and that the users get what they deserve for living in the stone age...

In contrast, the design goal of LISTSERV was to combine the power of the emerging network with that of the VM server concept, and build a tool that would show *ordinary* users what the network could do for them. Hopefully, more people would use the network and tell their managers how much time they saved thanks to the XYZ-L list, and there would be money for faster network lines and bigger budgets for the computing centres. And, while LISTSERV was only a small piece of the puzzle, it happened. Today we have fast international networks, fast LANs and fast computers on the desks of people who were at best sceptical about the usefulness of computers in 1985, and the expression "the power of the network" has actually become a *cliché*. We have the technology and credibility to build even faster networks, computers to serve growing numbers of users, and in fact things are moving so quickly now that we already have a *cliché* for all that - the *Information Super-highway*.

But our challenge for the remainder of this century is not the Super-highway, the exhaustion of IP address space or the commercialisation of the Internet. While these are serious issues of critical significance to the future of the Internet and large-scale networking, they are technical issues - something for computer experts, lawyers and economists to figure out while the users watch TV, just like thousands of lives are saved in hospitals every day while people watch TV. The real difficulty will be to build networks and tools that meet the *users'* expectations, and to teach our experts that, unlike networks and computers, the TV watchers have not changed much since the "stone age" of 1985. If our network experts fail to accommodate the simple needs of the Neanderthal man with all this

---

(1) Multipurpose Internet Mail Extensions, RFC1341

display of modern technology, other possibly less talented but more business-minded people will. For better or for worse, the personal computers and the "cavemen" are here to stay, and making their life more difficult than it needs to be would be a serious disservice to the Internet. This is why LISTSERV still lives in the stone age, and why all you will need in order to exploit its power is a copy of this handout, and the same basic understanding of e-mail as in 1985.

Contents

Foreword 1
What is 5
- Mailing lists - overview 5
- LISTSERV 6
- Filelists 6
- More about mailing lists 7
  - Public vs. private 7
  - Open vs. closed 7
  - Moderated list 7
  - Peered list 8
  - List header and keywords 8
- List owner, LISTSERV maintainer, postmaster, et al. - who does what? 9

How to 9
- How to send commands to LISTSERV 9
- How to upset a lot of people who don't even know you 10
- How to locate interesting lists 11
- How to join and leave a mailing list 11
- How to send mail to a list 12
  - Posting new messages 12
  - Replying to messages from the list 13
  - ALWAYS pause to consider where to send your reply! 13
- How to interpret mail headers 14
- How to see who is on the list 16
- How to access list archives 17
  - Using the INDEX command 17
  - Ordering archived postings with the GET command 17
  - The database functions 18
- How to access a filelist and use the file server functions 18
  - Ordering large files via mail 18
  - Ordering binary files 19
  - Subscribing to files 20
- How to review and customise your subscription 21
  - MAIL, DIGEST, INDEX and NOMAIL 21
  - REPRO and ACK 22
- How to deal with rude people 22
- How to get more information 24

The global view 24
- The automatically maintained "list of lists" 25
- The "Global List Exchange" (GLX) 26
  - Using the GLX to reach mailing lists 26
  - Using the GLX to order files 26
  - Using the GLX to contact list owners 27

Future plans for LISTSERV 27
- LISTSERV on VMS™, unix® and Windows NT™ 27
- Standardisation efforts 28
- SENDFILE for the Internet 29
- User-friendly interfaces for personal computers 29

What is...

Before we get to the "how to" chapter, it is necessary to introduce a number of basic LISTSERV concepts to avoid misunderstandings and confusion, especially as some of these words are sometimes erroneously used with a different meaning on the network.

What to read in this chapter

If you are new to LISTSERV, you will want to read at least the descriptions of "Mailing lists" and "LISTSERV" before you move on to the next chapter. If on the other hand you are already familiar with LISTSERV lists, you can skip these definitions and just come back to them as you encounter things you do not understand.

Mailing lists - overview

A **mailing list** is a list of people's names and addresses that is used to send certain messages or announcements to many people at once, who are usually expected to share a common interest in the contents of the message - just like in the real world. However, and unlike in the real world, you can usually join and leave the list as you see fit, so there are good chances that you will actually find at least some of the messages interesting. Furthermore, you can send your own messages to the mailing list at no cost. In fact, electronic mailing lists are more like clubs or magazines than a "real world" mailing list.

A mailing list is managed by a **list owner** (or sometimes several owners for large lists). The list owner is the person with formal responsibility for the operation of the list - some kind of referee, if you want. The list owner defines the list's charter and policy, *i.e.* what the list is about and what are the general rules all subscribers must accept in order to be allowed to join the list (don't worry - usually these rules are pretty lax). He is also responsible for all administrative matters and for answering questions from the list subscribers. It is not unusual to have several list owners spreading the work and responsibility among themselves; in particular, it is common for a "technical" list owner to assist the non-technical person who is formally in charge of the list with administrative matters.

The messages sent (or **posted**) to a mailing list are usually saved in the **list archives** for future reference, although this function can be disabled to save disk space. Other expressions you may encounter are **list notebooks** and **list logs**; they all mean the same thing to people from different computer cultures. These archives are usually organised in monthly **log files**, although high-volume lists may use weekly files and some lists use more sophisticated arrangements. A log file is just a disk file containing everything that was said on the list on a given month (or week). There are two ways to access these list archives: you can ask LISTSERV to send you (say) the log file for March 1993, or you can use the **database functions** to **search** the archives for messages related to a particular topic, or sent by a certain person, and have LISTSERV return a copy of the messages that matched your search criteria. The database functions take some time getting used to, because when you are searching an archive with 5000 messages it can be difficult to select the messages you are looking for without also selecting another 200 unrelated messages. But, once you get past that obstacle, they are invaluable.

Given a reliable and reasonably fast network, mailing lists are highly interactive. When you send a message to a mailing list, LISTSERV will distribute it immediately and you can expect most subscribers to receive their copy within **5 to 20 minutes**, depending on location and mail system. While this is great if you are looking for the answer to a problem your boss just reminded you he wanted solved yesterday, in some cases it can be annoying because you would rather not be interrupted while you are working. When you are not in a hurry, it can often be more convenient to read all these messages during a break, or at any other time where you are not too busy. LISTSERV provides **digest subscriptions** for this purpose (also called "list digests" or "subscriptions in digest format"). A digest is simply a larger file with everything that was said on the list in a particular day (or week for low-volume lists). Unlike their real world counterparts, LISTSERV digests are **not edited** and you get exactly the

same information as with a normal subscription, just in a single message that is usually sent during the night when the lines and computers are less busy. Occasionally, the amount of activity on the list may be so high that LISTSERV will send a **special issue**, to prevent the next day's digest from becoming too large (many mail systems reject messages larger than 100 kilobytes, and some PC mail programs and editors cannot view messages larger than 64k). This special issue is sent immediately, when LISTSERV decides the list has become hyperactive.

Sometimes even digests take up too much of your time to be worth the effort - or maybe you have a limited amount of disk space to store your mail, and the large digest files occasionally fill up your disk or quota. If your interest in the list is only peripheral, you may want to get an **index subscription**, which is similar to a digest but much smaller, as it only contains a directory of all the messages posted in the last day. That is, LISTSERV sends you a list of all the messages, in chronological order, with the name and address of the author of the message, the message subject, and its size (in lines). It only takes a few second to identify the messages you are interested in, and you can then order a copy of just these messages from LISTSERV. And if there was nothing interesting that day, all you have to do is throw the index away.

There are over 4,000 public mailing lists on BITNET, covering virtually all imaginable topics in academia (and then some). Think of it as a virtual encyclopaedia that is always up to date: no matter what you need to know, or at what time of the day, the chances of finding a list from which you can get an answer (usually in a matter of hours, or just minutes if you are familiar with the database functions) are very good. The more specialised your questions, the higher the chances of success. You must, however, find the right list, and we will see later how LISTSERV can help you with that.

LISTSERV

LISTSERV is the program that does everything we will be talking about in this document. The first version was written in 1986 by Eric Thomas under the name *Revised LISTSERV*. Nowadays it is a commercial product, distributed by a company called *L-Soft international*. LISTSERV is always spelled in upper case. You can of course type the name in lower case when sending commands, but this can be confusing when you are writing to other people. For instance, some people say "a listserv" when they really mean "a mailing list", as in "I am looking for a listserv on biochemistry". This is about as appropriate as saying "I am looking for a wordprocessor on the Stockholm syndrome", when what you really mean is that you are looking for a paper or thesis on that syndrome. And, of course, technical people will think you are actually looking for a word processor, and send you a lot of technical information you neither need nor understand.

Another common misconception is that "listserv" is a generic English word, like "electronic". Some people say "a listserv list" indifferently for a "real" LISTSERV list and a list managed by a totally different and incompatible mailing list manager. This is confusing because people will then think the list is managed by LISTSERV, and assume certain functions are available and commands are sent in a certain way. For instance, they may assume they can join the list without having to know where it is located (this will be explained later on), and complain that it does not work. Now, this may sound like nit-picking and an unfair imposition on the memory of non-technical users, but on the other hand people have no trouble remembering that "a Mac" is not an acceptable way to refer to an IBM® PC. Just think of LISTSERV as a trademark for a particular brand of mailing list manager, and you will avoid all these mistakes.

Filelists

In addition to mailing lists, LISTSERV also acts as a **file server** - a program that manages collections of files and makes them available to users upon request. Among these files are the list archives we have already mentioned, but LISTSERV can store just about any kind of file: papers put up for discussion, agendas and minutes of upcoming meetings, survey results, programs, *etc*. These files are organised in **filelists**, which are very much like directories on a PC. Each filelist contains a list of files, along with some descriptive text and two **file access codes** (or **FAC**) that define who is allowed to order a copy of the file and who is the person in charge of updating it (the so-called

file owner). List archive files are owned by the list owner, file access code OWN. Public files have file access code ALL whereas files only available to people subscribed to the list are marked PRV (private). There are many other codes but ALL, PRV and OWN is all you need to know about initially.

If this is your first time through this document, you may want to jump to the "How to..." section now and come back to the following definitions later, or as you encounter words you are not familiar with.

More about mailing lists

Public vs. private

A **public** list is a totally open list - anyone can join or leave, ask questions, see who is on the list, search archived messages, and so on. Public lists usually attract a lot of subscribers, and tend to generate quite a lot of traffic. They are ideal candidates for digest subscription (see later on). Sometimes two people decide to have an argument on the list and your mailbox is flooded with repetitive and pointless messages the next morning; if that happens, just wait for the list owner (see below) to take action or, if you don't have the patience, sign off the list and wait a couple days before subscribing again.

Conversely, a **private** list is a list exercising some measure of access control. Usually, you need to apply for membership to the list owner, and only people who are subscribed to the list may send messages and access archived postings, but there are many other possibilities. Private lists are usually smaller, more focused, and more "professional".

Some lists do require you to apply for membership but are still called "public". Usually this is because the list owner really lets anyone join the list, but has to ask a few questions before letting you through, for instance because the list is funded by a grant which requires every participant to state which organisation he is working for.

Open vs. closed

In most cases, open/closed are interchangeable with public/private; they are mentioned mostly so that you do not get confused the first time you encounter these expressions. An **open** list is one you can join and leave as you want; with a **closed** list, you have to apply to the list owner for membership. If a list is operating with **closed subscription**, it means nobody can join at all: this is typically used for lists corresponding to staff, boards of directors or other formal structures one cannot apply for via e-mail.

Moderated list

With a normal list, messages submitted to the list by users are either accepted or rejected. If the message is accepted, the original text is published in its entirety, and the other subscribers can know that nothing was censored. A **moderated** list is similar to a real-world newspaper: when you send mail to the list, it is opened by a human being, called the **editor** or the **moderator**, who then decides what to do with your message. Usually the editor "cleans up" your message, shortens it if it was too long, and includes it in the next "issue" (this is called a "moderated digest", because the editor sends a new issue at regular intervals with selected contributions from the readership). Sometimes the editor just acts as a filter, deciding whether or not to accept articles.

Peered list

A **peered** list is a list that runs in parallel on multiple computers, both to spread the load between several machines and to provide better reliability (you can still access part of the list even if one of the machines is down). A peered list is in fact a collection of individual lists, called **peers**, which all receive and deliver exactly the same messages. You can send your messages to any of the peers, and they will be propagated to the others. There is no "master" list: all the peers are equivalent. When you subscribe to a peered list, your request is forwarded to the peer that is nearest to you.

Peers are now mostly a historical artefact. Since the inception of the load-balancing algorithm known as DISTRIBUTE, in 1987, peers are no longer useful to reduce network load. On the other hand, for high-volume lists it is always a good thing to keep the archived postings and associated documents on more than one system, so that they remain available should a province or two be struck by a major natural disaster. Since peers are the best way to maintain two copies of this information in parallel, they are still in use for the largest lists.

List header and keywords

All the options you have seen are defined by **keywords** in a special section of the list, called the **list header**. When you examine the list with the REVIEW command (which will be described later), the list header is at the top, before the list of addresses and names; all the lines in the header start with an asterisk. The list header contains the **title** of the list, which is a one-line description of what the list is about, a number of keywords, and usually some additional information about the list itself. The reason you need to know about all that is that there is no way to ask LISTSERV "So, is the XYZ list public or private?". While it is easy to write a program to answer that particular question, there are 45 possible keywords, many of which are related to each other, and you will probably want more information if the program just answers that the list is private - who exactly can post, who can subscribe, *etc*. You quickly reach a situation where it is not much easier to remember the syntax for asking the questions than to learn the "list header language" and answer your own questions better than any computer could. Here is a typical list header:

```
*
* LISTSERV give-and-take forum
*
* Review= Public    Subscription= Open       Send= Public
* Notify= Yes       Reply-to= List,Respect   Files= No
* Stats=  Normal,Private
* Notebook= Yes,L,Monthly,Public
* Peers=  UGA,SEARN
*
* Owner=  ERIC@SEARN       (Eric Thomas)
* Owner=  Quiet:
* Owner=  HAROLD@UGA       (Harold C. Pritchett)
*
```

Some of the keywords may look pretty daunting, but they are usually the ones you do need to know about. Most of them are in fact fairly intuitive: "Send= Public" means anyone can send messages to the list, "Subscription= By owner" means that you have to go through the list owner to subscribe to the list, and so on. Note that, in the list header language, the word "Private" has a more specific meaning than in the rest of this document; it means "all the people currently subscribed to the list". Thus, "Send= Private" does not mean that the list is private rather than public (although that is actually true, it is a simple observation rather than a definition), but that only people who are subscribed to the list can send to it. "Postmaster", if you ever encounter it, refers to the people who manage the LISTSERV software on that machine.

List owner, LISTSERV maintainer, postmaster, *et al.* - who does what?

The **list owner**, as explained above, is the person formally in charge of the operation of the list. The list owner is usually an expert in the field covered by the list, but he may not know much about computers, and he often does not work in the department or organisation where the list is physically located (this is called a "remote" list owner). The responsibility of the list owner is limited to the list itself, and does not include the computer running the list, its mail system, network lines, *etc*. The **LISTSERV maintainer**, on the other hand, is the technical person in charge of the LISTSERV program on a particular computer. While he may not be in charge of the entire computer, he will usually work where the computer is located and know where to get help if there is a problem he cannot solve on his own. The LISTSERV maintainer creates new lists and allocates disk space to store information related to the list, but often does not even know what the list is about. As a rule of thumb, you should never contact the LISTSERV maintainer unless you are sure that the problem you are having is due to LISTSERV itself. Remember, the list owner usually has only one list to take care of, whereas the LISTSERV maintainer oversees dozens of different lists and, quite understandably, tends to think that a "good" list is one giving him no hassle, all other factors being irrelevant.

In technical discussions, the LISTSERV maintainer is often called "the postmaster", because this is how the corresponding line in the LISTSERV configuration files reads. In 1986, the LISTSERV maintainer was usually the person in charge of the mail systems, which is commonly called the "postmaster" and receives mail sent to the special address by the same name. As computer networks grew, the postmaster task became more than enough work for a single person, and large universities now typically have one person answering mail sent to the special *postmaster* address, another one maintaining the mail system, and yet a third one taking care of LISTSERV. The "genuine" postmaster is the person to contact if you have questions about the university he works for, or if you have problems with the mail system or gateway. If the list owner is away, it is a better person to contact than the LISTSERV maintainer: in the worst case it will be the same person, and if you are lucky it will be someone paid to answer questions from normal users without getting upset.

How to...

In this section, we will see how to use LISTSERV to locate a mailing list related to a particular topic, subscribe to it, participate in the discussions, and otherwise "get information out of the system". To make it easier for computer experts to understand this tutorial, we will choose a topic with which they are undoubtedly familiar: *the influence of beta-casein in the proteolytic system of lactobacilli.* All the examples in this tutorial are real.

Some instructions are only available to users with direct BITNET connectivity (*i.e.* people who can send mail to BITNET without having to go through a gateway). These instructions may be confusing to other users and have been marked like this:

> *BITNET:* You can also send commands to LISTSERV via interactive messages.

If you do not have BITNET connectivity, there is no need to read these instructions.

How to send commands to LISTSERV

To send a command to LISTSERV, you will already need to solve a problem whose existence probably hasn't occurred to the network experts:

Where do I find a LISTSERV to send commands to???

Well, the best solution is to use the LISTSERV at your own university, if there is one and you know on what machine it is located. But if you don't know or you are not sure, just use the address:

LISTSERV@LISTSERV.NET

All you have to do is to send mail to LISTSERV@LISTSERV.NET, or maybe to your organisation's local LISTSERV if you know its address, and type the commands you want it to execute in the ***text*** of your message. LISTSERV ignores what you put in the subject line, so it doesn't matter if your mail system inserts NO SUBJECT, INTER-OFFICE MEMORANDUM or some other nonsense. You can enter several commands as long as you put them on separate lines. When you are done, send the message and LISTSERV will send you a reply by mail.

BITNET: You can also send commands to LISTSERV via interactive messages (TELL on VM, SEND on VMS™). In that case the generic address is LISTSERV@LISTSERV. The special node LISTSERV is a generic node, like INTERBIT, which can correspond to any machine offering the service in question.

How to upset a lot of people who don't even know you

All tutorials have their share of pedantic, boring statements in bold letters that have an irritating tendency to insinuate that you can't read, and this one is no exception. So here goes the necessary evil:

If you want to send a message to all the PEOPLE on the mailing list, the right address to use is:

listname@hostname
(example: SFS-L@SEARN.SUNET.SE)

If you want to send a COMMAND for the computer to execute, the right address is:

LISTSERV@hostname
(example: LISTSERV@SEARN.SUNET.SE)

If you want to contact the PERSON who manages the mailing list, write to:

listname-Request@hostname
(example: SFS-L-Request@SEARN.SUNET.SE)

Since we are delivering admonitions, a couple notes for the purists are in order. You will probably want to skip them on first reading.

A few mailing list managers do accept commands sent to the *listname-Request* address, and some widely advertise this as yet another step away from the stone age. However, this is a very controversial use of an age-old Internet convention that says that you write to *listname-Request* if you have any question about the list in question. While it doesn't say you cannot send a SUBSCRIBE command to this address, list managers did not exist at the time this convention was introduced. What is clear however is that you are entitled to a better answer than *"Unknown command: PLEASE"* if you send mail to that address in English. The reason for this storm in a teacup is that technical people tend to get upset if you suggest that they are not capable of writing a program that can reliably tell commands (that may contain errors) from messages in human languages (which may or may not be English). No matter what you may have heard, the LISTSERV address always works for commands, and there is nothing to be gained by sending them to the *listname-Request* address instead.

LISTSERV also implements a *listname-Server* mailbox, which is a synonym for LISTSERV. Since it is a relatively recent addition, it is safer to send your commands to LISTSERV. The *listname-Server* convention is mostly useful

for the GLX, which will be described later (it lets you use a list or send commands to the corresponding LISTSERV without having to know where the list is physically located).

How to locate interesting lists

Now that you know how to send commands to LISTSERV, the next thing to do is to locate a list with the right kind of experts to answer your question, which we will repeat here for the benefit of the feeble-minded computer experts who, just for a change, forgot to write it down when we told them and are now coming back to waste everyone's time with the same stupid questions. Your task is to find out:

What is the influence of beta-casein in the proteolytic system of lactobacilli?

The simplest way to look for a list is to search the so-called "list of lists" that LISTSERV maintains automatically. Now, it would probably be unrealistic to expect to find a list dedicated to the influence of beta-casein in the proteolytic system of lactobacilli. Even with over 4,000 lists, that would still leave us with a couple dozen lists on various aspects of lactobacilli, and not very much space left for the computer experts to talk about virtual reality and C++. In fact, a list about lactobacilli seems like the most likely place to find the answer. So let's look for that and see what we find.

The problem at this point is to find a way to bypass the computer's innate stupidity. If we ask for *lactobacilli* and what it had in its list was *lacto-bacillus* or *bacillus lactis* or even just *lactid acid,* it is liable to say there is no such list and we will have to spend 15 minutes trying all possibilities. The simplest thing to do is to request two searches on the root words, *i.e.* one on LACT and one on BACILL. This should catch most related lists, even though it will probably return more lists than wanted. So let's send mail to LISTSERV@LISTSERV.NET with the following two lines in the message text:

```
list global bacill
list global lact
```

Unfortunately, after executing the first command LISTSERV claimed there was no list about bacilli. It did return something for the second command, though:

```
Excerpt from the LISTSERV lists known to LISTSERV@SEARN on 31 Aug 1993
Search string: LACT

Network-wide ID  Full address      List title
---------------  ------------      ----------
GALACTIC         GALACTIC@UGA      GALACTIC Industries Discussion
LACTACID         LACTACID@SEARN    Lactic Acid Bacteria Forum
```

Bacteria, of course! It would have been a good idea to do a search on that as well (this would have shown LACTACID and another list, about cyanobacterial toxins). Anyway, this LACTACID list looks like it is about lactobacilli, so the most difficult has been done and all we need to do now is join the list and ask our question.

How to join and leave a mailing list

Given the list name, joining the list is very easy. The command that needs to be sent is:

```
subscribe lactacid
```

And as usual, the command can be sent to LISTSERV@LISTSERV.NET or any other LISTSERV. At this point, you can run into two minor problems if you are unlucky:

- LISTSERV insists on having a "real world" name for everyone on the list, because addresses such as 00038385@XXXMAIL.COM are not very informative to the other users of the list. Normally, LISTSERV will extract both your e-mail address and your real name from the mail headers generated by your computer. But some computer programs do not supply any real name, and LISTSERV may ask you to resend the command with your name, as in:

  subscribe lactacid Anna Galiena

  LISTSERV will then automatically insert the name you supplied in the headers of messages you send out to the list. When other people read the messages you post, they will see your name in addition to your e-mail address, and will not have to strain their memory with meaningless sequences of numbers.

- While you will never have this problem with lists you found by searching the "list of lists", some lists are confidential or otherwise not globally known throughout the LISTSERV network. In such cases, you must send the subscription to the server that is hosting the list, because the other servers (and in particular LISTSERV@LISTSERV.NET) will not know where to forward your request.

BITNET: If you send your subscription request via interactive message, LISTSERV has no mail header to extract your name from and you must type it after the name of the list. If LISTSERV already knows your name from a previous subscription request, it is enough to type the name of the list.

When you join a list, you are sent a little pamphlet which looks very boring and does not seem to have anything interesting to say about the list itself. Do not discard it! Treat it like a warranty card - no immediate value, but you never know when you might need it. You should make a new folder in your mail program for these little pamphlets and search it whenever you have an administrative question about a mailing list, instead of contacting the list owner directly. Although they do look the same from a distance, the pamphlets are customised to the individual lists and do not contain the same information. Finally, saving them in a dedicated mail folder makes it very easy for you to know what mailing lists you are subscribed to, and when you joined.

You can leave the list at any time by sending a SIGNOFF command:

signoff lactacid

LISTSERV does not need your name since you are leaving anyway, so there is no need to type it. Do not hesitate to subscribe to a list to see what it is really about, and then sign off a couple days later if it turns out not to be what you expected. Most list owners are used to that and will not be hurt or angry.

How to send mail to a list

Now that we have subscribed to the list, all we need to do is ask the question and wait for someone to answer. To avoid wasting the time of the LACTACID subscribers with a fake question, no "real" example will be shown here. The answers would probably be too long to reprint anyway.

Posting new messages

To post a new message to the list, all you have to do is send mail to the "list address", using the same procedure as when you send mail to other people. Your mail program does not need to know that you are sending to a list. The list address is the name of the list, followed by the name of the machine where it is hosted, in our example:

LACTACID@SEARN.SUNET.SE

You can also use the GLX if you forgot where the list is located. This will be explained later.

Depending on how the list is set up, LISTSERV may or may not send you a copy of the messages you post. No matter which behaviour the list owner chose to have by default, you can always instruct LISTSERV to behave the way *you* want it to. This, too, will be explained later.

Replying to messages from the list

Once you become familiar with LISTSERV and mailing lists, you will probably want to return the favour and help other people when they ask questions that fall in your field of expertise. Don't be shy! People will not get upset at you for trying to help. On the other hand, people do get upset when someone consistently gives incorrect answers, especially if this is not a genuine mistake but something like "Well, I am not sure but I heard a colleague saying he read on a paper that someone claims that..." As a rule of thumb, you should answer questions if:

- You know the answer for a fact, and not from hearsay.

- The question has not been answered by 5 other people before you. It can be annoying to see the same answer over and over. The simpler the question, the more likely it is to have been answered by someone before you. So when the question seems easy, it is a good idea to finish reading all your mail before replying, so that you can see if someone else has already answered.

The best way to answer a question is by using the "reply" function of your mail program (which is sometimes called "answer", "respond" or something similar). This way the message subject is preserved and the other subscribers can see that your message is a reply to the original question. You can of course post a new message, but you will then have to retype the subject, and if you enter something slightly different people may not realise it is a reply to that question.

ALWAYS pause to consider where to send your reply!

There is no universally correct place to send your reply. Most of the time, your reply will be useful to at least one other person on the list (beyond the one who asked), but on the other hand that might be only a small fraction of the list membership, and some people might complain that you are wasting their time (some people say that anyway, so don't worry unless several people seem to share this opinion). In general, if your reply is short there is little harm in sending it to everyone: it does not take much time to discard a message which is not interesting. On the other hand, if your reply is a 2000-lines paper you wrote on the subject it might not be a good idea to send it to the list unless you are sure everyone is interested. Some people have to pay for mail by the byte, or to download it to a personal computer through a low-speed modem. The best thing to do in that case is to send a short message to the list saying you wrote this paper, and that people who are interested can contact you for a copy. If you find 50 requests in your mailbox the next morning, contact the list owner and suggest that he make the paper available from LISTSERV, so that subscribers who are interested can order it directly from the server and you are not interrupted every 15 minutes with a new request.

But most of the time, people will just click on the "reply" icon (or press the "reply" function key, or whatever) and let "the system" send the message wherever it thinks it should go. Usually this is because people are busy and don't have time to learn the subtleties of electronic mail, or then because they just hit the reply button out of habit, without even thinking about it. Most lists are organised as "forums" where public discussion is actively encouraged, and since "the system" will usually send the reply back to the list, the reply button usually does what people want and they have no incentive to learn about the various reply options of their mail program. Unfortunately, this can sometimes be embarrassing if you end up inadvertently sending a private comment to the whole list.

Fortunately, there is a very easy way to avoid this, and the good thing is that it works even for non-LISTSERV mail. It is a simple rule that is easy to remember once you understand its purpose:

Always pause before sending a message you wouldn't want quoted against you!

And ask yourself a few simple questions:

- Who is getting the message? Carefully check who your mail program intends to send the message to, and make sure this is where you wanted it to go. It is easy to click on the wrong icon, press the wrong key, misunderstand the meaning of a help file, or otherwise do something silly that will make your computer send the message to the wrong people.

- How well do you know these people? Can you trust people you have never met in person not to forward your comments to someone else, or to a list? And if they did, whose reputation would suffer the most - yours for saying all these mean things, or theirs for forwarding without your permission?

- What is the worst thing that can happen to you if this message is used against you? Computers are not perfect and they sometimes do unpredictable things to perfectly valid messages. It may be a rare occurrence, but it happens; any system manager will have a lot of juicy stories to tell you about messages that were forwarded to him because they caused some system problem or other, and whose contents could have made a couple people lose their job if it had been shown to the right person. System managers are supposed to have a code of ethics, but do you really want to rely on that?

You want to ask yourself these questions anyway, even if the message has nothing to do with LISTSERV, even if the list is set up to reply privately by default. In a non-computer situation, you would probably look around to see if someone can overhear you. Just use that same reflex to look around the list of recipients and decide if you can trust these people with what you said. If you develop this habit, you will never send to a list by mistake.

How to interpret mail headers

Now that you have subscribed to the list and asked your question, people are going to start replying to you. They will probably send their answers to the list, since it is a question that might interest other subscribers as well, and you will start receiving your first messages from the mailing list. It can be a bit confusing the first time, depending on the kind of mail program you are using. A typical message from a LISTSERV mailing list might look like this:

```
Date:       Wed, 1 Sep 93 16:53:27 IST
Reply-To:   EARN Group on Information Services <EARNINFO@EARNCC.BITNET>
Sender:     EARN Group on Information Services <EARNINFO@EARNCC.BITNET>
From:       David Sitman <A79@TAUNIVM.TAU.AC.IL>
Subject:    GNRT2 - call for comments
To:         Multiple recipients of <EARNINFO@EARNCC.BITNET>

The EARN Staff is now finalising the text for the second edition of
the Guide to Network Resource Tools. (...)
```

The date and subject lines have the same meaning as with normal mail messages. The "Sender" line tells you what *list* the message is coming from, whereas the "From" line contains the name and address of the *author* of the message. The "Reply-To" line shows where your reply will go if you just click on the reply icon without explicitly telling your mail program what to do. Sometimes there is an "Organization" line that tells you where the author of the message works (LISTSERV cannot know that, so it will not be present unless the author's mail program provided it). At any rate, this is called a "short header": it shows you all the important information you need to know, and nothing else.

Depending on your mail program, this message can look like it came from either EARNINFO@EARNCC or A79@TAUNIVM.TAU.AC.IL when you check the list of new messages in your mailbox. If you only subscribe to one mailing list it does not really matter, but once you start participating in several lists you will quickly realise that it is a lot more convenient to have messages from mailing lists shown under the list address rather than the author's address. Unfortunately, LISTSERV cannot help you here: each mail program is different, and there is no standard way of telling mail programs "this is what you must show in the mail directory". Luckily, most mail program do show the list address, so you will probably not need to do anything. Many mail programs can be configured to show either the list address or the author's address, and your user support staff may be able to help.

Unfortunately, there are also many mail programs which cannot be made to show the list address no matter how carefully you read the manual, and which in fact do not even show you the mail header when you open the message to read it. That is, when you double-click on the message from the directory, what you see starts with the sentence about the second edition of the Guide to Network Resource Tools, and there is no way to see the "Sender" or "From" lines at all. In other words, the program either shows you who sent the message or what list it came from, but not both, and, no matter how many options you try, it will never show you both because the information has been removed by the gateway that connects your LAN to the Internet. While the right thing to do is to complain to the company that sells this software, sometimes this is just a lost fight, as they answer that you do not need to see these lines, and that it would be too complicated to explain why since you don't know anything about computers, whereas they do. Here LISTSERV can help you by sending "dual headers", which contain a second copy of the important fields from the mail header (name of the list, author, subject, *etc*.) To activate this option, you would send the following command to the LISTSERV address:

```
set earninfo dualhdr
```

You will still not be able to see that information from the mail directory, but once you double-click on the message to read it, it will be there right at the top of the message:

```
------------------- Information from the mail header --------------------
Sender:     EARN Group on Information Services <EARNINFO@EARNCC.BITNET>
Poster:     David Sitman <A79@TAUNIVM.TAU.AC.IL>
Subject:    GNRT2 - call for comments
-------------------------------------------------------------------------
```

Finally, some mailing lists may send messages with very different headers, like this:

```
Return-Path: <@SEARN.SUNET.SE:owner-novell@SUVM.SYR.EDU>
Received: from SUVM (NJE origin BITMAIL@SUVM) by SEARN.SUNET.SE (LMail
          V1.2a/1.8a) with BSMTP id 3968; Sun, 5 Sep 1993 11:16:22 +0200
Received: from SUVM.SYR.EDU by SUVM (Mailer R2.10 ptf000) with BSMTP id
          7184; Sun, 05 Sep 93 05:18:11 LCL
Received: from SUVM.BITNET by SUVM.SYR.EDU (LISTSERV release 1.7f) with
          NJE id 4541 for NOVELL@SUVM.SYR.EDU; Sun, 5 Sep 1993 05:17:58
          -0400
Received: from SUVM by SUVM (Mailer R2.10 ptf000) with BSMTP id 7176;
          Sun, 05 Sep 93 05:16:03 LCL
Received: from isgate.is by SUVM.SYR.EDU (IBM VM SMTP V2R2) with TCP;
          Sun, 05 Sep 93 05:16:01 LCL
Received: from rfisk.is by isgate.is (5.65c8/ISnet/14-10-91); Sun,
          5 Sep 1993 09:15:08 GMT
Received: by rfisk.is (5.0/ISnet/11-02-92); Sun, 5 Sep 93 09:15:05 GMT
Message-Id: <9309050915.AA01846@rfisk.is>
X-Mailer: ELM [version 2.4 PL21]
Mime-Version: 1.0
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 8bit
Content-Length: 1594
```

```
X-Charset: ASCII
X-Char-Esc: 29
From: root@rfisk.is (Super User)
Subject: Strange server behavior
To: novell@suvm.syr.edu
Date: Sun, 5 Sep 1993 09:15:04 +0000 (WET)
Cc: root@rfisk.is (Super User)
Sender: owner-novell@suvm.syr.edu

Hi, (...)
```

This is because LISTSERV "short headers" are politically incorrect, and experts who know what is good for you would like to see them totally banned and replaced with this. Since LISTSERV's philosophy is to adapt to the users as much as possible, rather than require them to adapt to the computer, and since computer experts are also users, after all, there is an option to give them the type of headers they like. If you subscribe to a mailing list managed by a computer expert, chances are that he will have set up the list so that you get this type of header by default, since he knows they are better for you. If that makes your life more difficult, all you have to do is send the following command to LISTSERV:

```
set earninfo short
```

The main reason why some computer experts get so upset about short headers is that they are convinced they create vicious mailing loops. This is a myth rooted in some of the details of the convoluted histories of BITNET and LISTSERV, which would be too long to explain here. If you ever get into a fight with a local computer expert, you may want to bring the following to his attention, so that he can revise his judgment:

- LISTSERV *always* sets the SMTP return address to *owner-listname* if the local mailer supports it - even for short and dual headers. Delivery errors are *not* normally sent to the list mailbox. *In addition,* LISTSERV has a state-of-the-art loop detector that hardly one error message in 10,000 can fool. Should the loop detector fail, there are a number of additional safety mechanisms to prevent the loop from iterating indefinitely; in the very worst case, the list will have generated as much traffic as during a busy day.

- Most mailing loops nowadays are caused by mailing list managers other than LISTSERV, which do not have a loop detector (or whose loop detector is not sophisticated enough). Even list managers which do not use short headers are vulnerable. In fact, they are the most exposed as they usually assume that nothing bad can happen to them; when it does, they are totally unprepared and vulnerable.

- LISTSERV delivers about 100 million messages a month. One of these messages, on the average, starts a mailing loop which causes at most 50 unwanted messages to be distributed to one list. The impact on the network is virtually nonexistent.

In other words, mailing loops are simply not an issue.

How to see who is on the list

To see who is subscribed to the list, send the following command to LISTSERV:

```
review lactacid
```

LISTSERV will return a copy of the "list header" and a list of all the subscribers. The list header contains the title of the list, various configuration parameters, and a short description of what the list is about (refer to the "What is..." section for more information about list headers). There are also some statistics about the list, after the name and address of the last subscriber, and there may be a mention of "concealed" subscribers. These are people who are on the list, but told LISTSERV not to tell anyone about them. Only the system managers and the list owner can find

out that they are on the list, unless of course they regularly reply to messages posted to the list or otherwise make it obvious that they are subscribed. While this concealment upsets many new subscribers, there is little or no point in starting a debate about this on the list: one of the reasons this option was implemented is that some countries have laws that require computer systems to protect the identity of users, should they so desire. While it would not be illegal to sell or run LISTSERV in these countries, running a mailing list without that capability might require prior consent from the subscribers and, the legal value of electronic mail being what it is, nobody wants to get into this kind of problem.

By default, the list of subscribers will be sorted by host name, *i.e.* people whose account is on the same machine will be next to each other. This is mostly for historical reasons (some list owners have programs that require the list to be in that order). You can also ask LISTSERV to sort the list by surname:

```
review lactacid by name
```

Sometimes, it may be more interesting to have the subscribers grouped by country:

```
review lactacid by country
```

Note however that LISTSERV has no way of knowing in what country people are actually living. All it can do is check the country in which their computer is located.

How to access list archives

Most LISTSERV lists archive everything that is said on the list for future reference, although many list owners regularly "trim" the archives to save disk space. As mentioned in the "What is..." section, list archives can be accessed in two ways: you can ask LISTSERV to send you the log file for a particular month, or you can use the database functions to search the archives for messages related to a certain topic, and have LISTSERV return a copy of all the messages matching your requirements. The first method is easier, but the second is more powerful.

Using the INDEX command

In order to find out what archive files are available for the LACTACID list, you would send the following command to LISTSERV@SEARN.SUNET.SE:

```
index lactacid
```

LISTSERV returns a file containing all the documents related to the LACTACID list, if any, and a list of the available archive files (always at the bottom, since people who are on the list are usually more interested in the other files, and the list of archive files can be quite long). Archive files have names such as LACTACID LOG9302. Sometimes a month is missing; this can happen if nothing at all was said during that month, or if the list owner decided nothing interesting was said and just deleted the file to save space.

Ordering archived postings with the GET command

To order a copy of an archive file, simply send a GET command followed by the name of the file:

```
get lactacid log9302
```

You can send several GET commands in the same message, as long as you put each new command on a separate line. Note that most LISTSERV sites will not let you order more than one megabyte of data a day (which corresponds to 15,000-20,000 lines of text). This is to prevent novice users from inadvertently

ordering more data than their mail server can handle, and then having to apologise to all their colleagues for having blocked the mail system for an entire day.

BITNET: The best way to send a GET request is via interactive message. This way you are sure that LISTSERV will return the file via SENDFILE, and not as a mail message that can be damaged by gateways running old software. This is particularly important if you are using 8-bit ASCII, which many mail systems do not support.

The database functions

As mentioned above, the database functions are more complicated and cannot be explained in 10 lines. For more information about the database functions, send an INFO DATABASE command to LISTSERV and you will be sent the documentation for the database functions. To give you an idea of what can be done, here is a quick preview:

```
Connecting to LISTSERV@SEARN, please be patient.

Welcome to LISTSERV@SEARN - Release 1.8a, backbone server.
CPU model 9221, DASD model 3380.

Enter command, or "QUIT" to exit:
search beta casein lacto bacill in lactacid
Search started...
--> Database LACTACID, 3 hits.

Enter command, or "QUIT" to exit:
index
Item #   Date    Time  Recs   Subject
------   ----    ----  ----   -------
000225 93/05/03 08:01   72   Program
000259 93/05/16 14:09  391   P7: Daniele Atlan et al (France)
000275 93/05/24 08:33  226   P11: L.STEPANIAK & P.F.FOX (Norway &
Ireland)
```

As you can see, the database functions let you find the answer to many questions without having to disturb the people on the list, and without having to wait for them to find the time to compose an answer of several hundred lines.

How to access a filelist and use the file server functions

The INDEX and GET commands described in the previous section also work for non-archive files. That is, if you know that a file called XYZ PROPOSAL is available from a particular LISTSERV, you can order a copy by sending a GET XYZ PROPOSAL command to that LISTSERV. Similarly, if you know that there is a filelist called PCSOFT on LISTSERV@ABC.EDU, you can order a copy by sending an INDEX PCSOFT command to that LISTSERV.

Ordering large files via mail

Most mail systems impose a limit on the size of mail messages, for technical or administrative reasons. Usually, "large" messages are simply rejected, even if you had enough disk space to store the message... and you are not even notified! So if you order a "large" file from LISTSERV, it may be dropped on the way and you will get the impression that LISTSERV did not send you anything. Unfortunately, there is no agreement as to what "large" means, nor is there any way for LISTSERV to find out what the various

gateways and mail systems between you and LISTSERV will accept. Or rather, the only way to find out is trial and error.

Generally speaking, the mail systems on most LISTSERV sites accept messages of up to 2-5 megabytes, which is usually enough (one megabyte being 15-20,000 lines of text). However, most mail gateways will reject anything larger than 100-200 kilobytes, and many personal computer gateways panic when they see a message larger than 20 kilobytes coming in. And 20 kilobytes is just a few hundred lines of text. Even with the so-called "network standard" limit of 100 kilobytes, you will be unable to order most academic papers or proposals.

When this happens, the first thing to do is to complain to your user support or system administrator, especially if you are one of the unfortunate victims of the 20 kilobytes limit. These ridiculously low limits are not just a waste of *your* time; they also waste the time of thousands of other people. The victims spend a lot of time trying to convince the rest of the world never to send them messages with more than 300 lines of text and never to prepare instruction files or agendas longer than that, while the rest of the world writes flames about the cost of disk space and computer time in 1993, and its relationship to maximum message size limits. Sometimes this limit is a political decision, to force the users of department A to use the computers department B (or computer manufacturer C) would like them to use. If you don't fight it, nothing will change and you will end up having to switch to a new computer system just to be able to get your work done.

Once you have duly complained, you can ask LISTSERV to split the file into a number of smaller files by adding a "SPLIT=" instruction to your GET command. For instance,

```
get lactacid log9305 split=100k
```

would instruct LISTSERV to send you a copy of the LACTACID logs for May 1993 in pieces of 100 kilobytes or less. When all the pieces have arrived, you can cut and paste them together to reconstruct the original file. A very tedious task, especially if you had to order several files and the various pieces arrive in random order, and yet another reason to complain to the powers that be. If you have a MIME-compliant mail program, it should be able to do this work for you automatically, although, depending on the program, you might be required to give special instructions to that effect.

Ordering binary files

Dealing with binary files can be a bit complicated, so you may want to skip this section until you are familiar enough with the network and computers.

Many of the files made available via LISTSERV are binary files, *i.e.* programs or word processor documents rather than plain text files. In order for these files to be useful, they must be transmitted using techniques which are compatible with binary files. Concretely what this means is that you cannot just mail a simple GET command and hope that the file will arrive in a usable form, because ASCII mail systems are not capable of transferring binary data directly.

BITNET: All you have to do is send your GET command with an interactive message, to make sure LISTSERV sends the file via BITNET and not as a mail message (if it knows your BITNET address, LISTSERV will send binary files via BITNET even if you order them via mail, but sometimes the address has not been registered by your system administrator). VMS™ users should use RECEIVE/BINARY if the file is meant to be downloaded to a personal computer, or is otherwise not a VMS™ program, BACKUP saveset, *etc*.

To order a binary file by mail, you have to ask LISTSERV to *encode* it before sending it. This encoding transforms the binary file into a text file, which can be mailed normally. When it arrives, you run a decoding program to return it to its original form. The most popular encoding format is called *uuencode*.

Unfortunately, it is also the least reliable. The characters used to represent the original data in the encoded mail file were chosen so that one or two computer instructions could be saved in the encoding and decoding programs, with the unsurprising result that many of them are not a very good choice and uuencoded messages occasionally get damaged when going through mail gateways. To solve this problem, *xxencode* was later developed; it works exactly the same way as uuencode, but only uses alphabetical characters, numbers and the plus and minus signs, which can survive all known gateways. Unfortunately, it failed to replace uuencode and it may be difficult to find decoders for this format. MIME's *base64* encoding is another variant that LISTSERV also supports; it is slightly more robust than xxencode, and has the advantage of being built-in to mail programs that supports the MIME mail extensions. In that case, LISTSERV uses the "Content-Type: application/octet-stream" attribute and refers to the format as MIME/APPL (there is another format called MIME/TEXT or just MIME, which corresponds to "Content-Type: text/plain").

At any rate, you have to choose an encoding format: UUENCODE, XXENCODE or MIME/APPL. LISTSERV cannot choose one for you because it has no way to know what decoding programs you have. You then add a "F=" keyword to your command, with the name of the format you chose, as in:

```
get winedit2a.zip f=uuencode
```

You can shorten the format names to UU, XX or MIME/A. And, of course, if the file is large you can also use the "SPLIT=" keyword described in the previous section.

Subscribing to files

Many of the files stored on LISTSERV are updated frequently. Software in general tends to be updated at regular intervals, and the same can be said about collections of files, such as agendas, minutes, newsletters, and so on. While the individual files may never change, new ones keep being added constantly. Checking for new files on a regular basis is a waste of your time, and just the kind of thing computers should be doing for people.

LISTSERV lets you subscribe to individual files or to collections of files. Two subscription methods are available: FUI (File Update Information) notifies you with a short mail message every time a file is changed, or a new file is added, whereas AFD (Automatic File Distribution) tells LISTSERV to send you a new copy right away. You should use FUI for files you are not sure you will want to order, otherwise AFD is probably more convenient. The two functions are independent, letting you subscribe to a file via both AFD and FUI if you want to (this can be useful if you want a copy of the file but also want to know the date and time of last update).

To subscribe to an individual file, you would send an AFD ADD or FUI ADD command, followed by the name of the file you are interested in; to subscribe to a collection of files, you would just use one or more asterisks in the file name, to define the collection you are interested in. For instance,

```
afd add bd* minutes
```

would set up an AFD subscription to the collection of files whose name matches the "pattern" you specified. For instance, the files called BD9305 MINUTES and BD9307 MINUTES would be part of the subscription, and so would the yet to be released BD9405 MINUTES.

You can check what files you are subscribed to by sending an AFD LIST (or FUI LIST) command. To unsubscribe from a particular file or group of files, use the AFD DEL (or FUI DEL) command, followed by the name or pattern of the files you no longer want to receive updates for.

How to review and customise your subscription

While the list owner defines the default behaviour of the mailing list, most of the parameters that directly affect the way LISTSERV communicates with you are under *your* control. That is, with the exception of options that would also affect other users, you can pretty much reconfigure the list the way you want it to behave.

Concretely, some "subscription options" are associated with your subscription to the list. When you join the list, they are set as defined by the list owner, but from then on you are free to update them. To see the current settings of your subscription options for the LACTACID list, you would send the following command to LISTSERV@SEARN.SUNET.SE:

```
query lactacid
```

And to change your options, you would use:

set lactacid *option1 option2...*

The new options you specify modify your old options, rather than completely replacing them. That is, you only need to type the names of the options you want to change. For instance, if you type neither CONCEAL nor NOCONCEAL, the concealment option remains unchanged.

There are too many options to give a comprehensive description of all of them here. Many of these options are pretty specialised anyway, and we will stick to the most important ones.

MAIL, DIGEST, INDEX and NOMAIL

In principle, after subscribing to the list you will receive messages from other subscribers immediately, as they are posted. This is the default behaviour of a mailing list, which corresponds to the MAIL option. If you would rather receive digests, for instance because you would rather not be interrupted while you are working, you should activate the DIGEST option:

```
set lactacid digest
```

Similarly, if you would rather receive a short index and order the individual messages you are interested in, the command would be:

```
set lactacid index
```

Note that some lists may not offer digests, either because they are running an old version of LISTSERV or because the list owner disabled them (this is very rare, though). Indexes are not available for lists which do not archive what is said on the list, because you would not be able to order the messages you are interested in. Finally, the frequency of digests and indexes (daily, weekly, monthly, *etc*.) is defined by the list owner. This is because, just like in the real world, digests are mass-mailed to people at regular intervals. It would be too expensive if everyone could decide when exactly and how often his private copy of the digest should be mailed.

When you switch from a digest or index to the MAIL option, LISTSERV immediately sends you a private copy of the day's unfinished digest or index, so that you can catch up on what has been sent to the list today. Otherwise you would never know what was said between the date of issue of the last digest you were sent, and the time you switched back to the MAIL option. This is a useful "trick" if you are curious about what has been said on the list today, and too impatient to wait for the daily digest. If you send the following two commands to LISTSERV, in the same mail message:

```
set lactacid mail
set lactacid digest
```

you will receive a copy of the day's unfinished digest, and you will still continue to receive digests normally as before.

Before going on vacations or leaving for an extended period of time, you may want to change your subscription to either INDEX or NOMAIL in order to avoid filling up your mailbox while you are not there to clean it up. If you switch to indexes, your mailbox will still have about one message from the list every day, but a reasonably short one, and you will be able to order the messages you are interested in when you come back. If even that would take up too much disc space, the NOMAIL option completely stops the flow of incoming mail messages. It is as though you were not on the list, except that you can restore your subscription when you come back by switching back to MAIL, INDEX or DIGEST. You do not have to subscribe again, which is particularly convenient on lists where you have to apply to the list owner for subscription.

REPRO and ACK

Some people like to get a copy of the messages they post to the mailing list, to make sure that they arrived in good shape and to see how fast they got distributed. Other people, on the other hand, find it particularly annoying to be interrupted by a copy of their own messages, especially if their mail program only says "New mail from the XYZ list" without pointing out that it is really their own message, and not the reply they are waiting for. This behaviour can be configured with the REPRO and NOREPRO options. When REPRO is active, you get a copy of your own messages, otherwise you don't. It can be a good idea to activate the REPRO option the first time you subscribe to a mailing list, to make sure the message arrives in good shape. Once you trust the system to deliver your messages properly and are no longer interested in checking them, you should probably turn it off.

Another option is available if you still want a short acknowledgement from LISTSERV every time you post something to the list: ACK. The advantage is that you still receive confirmation of the arrival and successful processing of your message, but the acknowledgement is always short (it does not include a copy of the message you sent) and does not fill up your mailbox if you often post large papers. It comes from LISTSERV rather than from the list, so you know it for what it is right from the "new mail" directory. The drawback, of course, is that you cannot check that your message did not get damaged on the way.

BITNET: When you choose the NOACK option, your postings will be confirmed by a short interactive message on your terminal. You can also use the MSGACK option, which provides the same information as ACK but via interactive messages.

How to deal with rude people

Help! I've just subscribed to a list about photography and asked a simple question, and everyone is insulting me! I don't understand why everyone is so rude!

Computer networks, just like the real world, have their share of rude people. While there isn't much one can do about it, it would be silly to avoid using the network (and, in particular, mailing lists) simply for fear that someone might insult you in public one day. Sooner or later, it will happen, and the best you can do is to be prepared for this. When it does happen, the only thing you absolutely must not do is whack the "reply" button and send off a stream of insults at your offender - or if you absolutely must, at least make sure that you do so in private. All you would achieve with a stream of insults is what is called a *flame war* in network jargon - dozens of people casting insults at each other, and a very swollen mailbox. Insulting someone on a public list is very much like punching someone in the face in a crowded bar near closing time; don't do it unless you want to get into a fight that could be painful for everyone.

Now, of course, you have been insulted and some factually incorrect statements may have been made about you, or your words may have been twisted around to make them sound like you meant exactly the opposite of what you said. A public reply may be appropriate, in much the same way that one would write to the editor of a newspaper and request the publication of a formal reply to "straighten out the facts". The important thing is to avoid content-free messages where no misinformation is corrected, no point is made and all that is ever exchanged is insults.

But before you do that, you will want to consider why these people have been rude to you. First of all, make sure the poster did intend to be rude. The network connects people from over 50 countries, and many of them are not native English speakers. They may have translated an idiomatic expression literally, and insulted you without meaning to. Similarly, native English speakers may have used a correct idiomatic expression which, when translated literally, sounds very mean in your language. The next thing to consider is where the poster comes from. No matter what your personal opinion on the question may be, there are cultures with a very different definition of what is or is not socially acceptable, and in particular there are cultures where *ad hominem* attacks are no big deal. While you may think that they should not do anything that hurts your feelings, you probably don't want to get into a cultural flame war, because you are probably hurting other people's feelings as well on a regular basis. For instance, do you always address people by their full name and title, or do you just say "As Peter said yesterday..."? In some countries, it is a grave insult to call people by their first name if you don't know them personally, while in others using the full title can sound sarcastic. There are dozens of similar examples, and the only way to successful cross-cultural communication is to tolerate other people's cultural habits in return for their tolerance of yours.

Now, if there is no possibility of cultural misunderstanding, you still have to dig a little bit further before hitting the reply button. Is there *any* valid reason for that person to be upset at you, setting aside the *manner* in which he vented his resentment? For instance, you could have asked a question whose inappropriateness should be evident even to someone without any knowledge in the corresponding field. A typical example is "What is the best camera I can buy for a reasonable price?" This is a so-called "stupid question", because other people have no way of knowing what you consider a "reasonable price", and there is no such thing as a "best camera" or a "best car" anyway - it all depends on what you want to do. If you think the complaint is legitimate and the only thing that upset you was the tone of the reply, it is best not to hit back. Bear in mind that most of the people on the list have a life and job outside of the mailing list, with all the frustrations and stress that we all have to go through now and then. Maybe that person had a bad day, or a lot of work to do, and was upset when he saw that his mailbox was crammed with "stupid questions". In particular, people whose job has nothing to do with data processing may experience chronic frustration when using their computer, especially if it runs programs designed by experts who "knew better" than them how they wanted the program to behave.

Which is not to say that you should be presumed guilty until proved innocent. While this may all have sounded like a plea for the right of rude people to insult just whomever they please, the key message was that you should think twice, and then twice again, before starting a flame war and causing hundreds of people to receive a pile of content-free messages. And if you presume yourself innocent of all charges until someone else proves you guilty on the list, this is exactly what will happen. The second point to consider is that, sometimes, people are having meaningful discussions in a tone that appears inappropriate to you, but that may seem perfectly normal to them. As long as their messages contain useful information, there is no point in trying to police the list, both because it is the list owner's job, not yours, and because adults are unlikely to change their behaviour in any significant way, especially if the people complaining are foreigners. If you want the list owner to take action, it is better to write to him directly, so that you do not end up being labelled as "one of the people running the flame war". If you just want to publicly express your indignation, it is best to type the message and pause, just before sending it, to consider whether you are doing this in the general interest or for your personal, selfish satisfaction. Most mail programs let you cancel a message if you have not sent it yet...

How to get more information

This document is much too short to cover all the aspects of LISTSERV. In particular, it says very little about the database functions, the file server functions, the various options at the list owner's disposal, LISTSERV's informational commands, and so on. You can order more information from LISTSERV, using the INFO command.

Important: To make sure that you receive up to date information, always send your INFO request to LISTSERV@LISTSERV.NET, unless of course what you want to know is the exact set of functions supported by a particular server. In that case you should send the INFO command to the server you are interested in.

The first thing you should order is the so-called "reference card", which contains a short syntax diagram for each command. It can be a good idea to print it and keep it at hand; it is ideal when you don't remember what option tells LISTSERV you want a copy of your own messages. The command you need to send to order a copy of this reference card is INFO REFCARD.

For more information about the database functions, use INFO DATABASE. For a list of all the documents you can order, send an INFO command with no parameter.

The EARN Association is also writing additional documentation for LISTSERV, as part of its documentation plan. The first of these documents is already available from the Association's file server, LISTSERV@EARNCC.EARN.NET (LISTSERV@EARNCC for BITNET users). To order a copy of this guide, send a GET LSVGUIDE PS command to this server, or GET LSVGUIDE MEMO for the plain text version.

The global view

One of the main reasons for the success of LISTSERV is that it was designed, from the very beginning, to be a ***distributed system*** with a ***global service view***. All right, nice words, but what do they mean in everyday English? Well, the "distributed system" bit means that several computers work together to provide the service. While we take this for granted in 1993 with client/server computing, it was not the normal way to get things done in 1986, at least not on mainframes. You were supposed to have a single central computer doing all the work, and indeed on BITNET we had a machine called BITNIC which ran all the mailing lists for the whole network. Needless to say, many people were upset when the link to that machine was down, and things could get *really* slow if someone started a flame war on a mailing list with 1000 subscribers. While part of the problem was that BITNIC was much too small a machine for its workload, when it came down to network load and single point of failure, there was clearly a flaw in the one-machine model. Today, the LISTSERV service is implemented by about 270 computers which deliver some 3-4 million messages a day on the average (on working days, and not counting digests and indexes). While a large mainframe can handle a tenth of that volume without problem, and with maybe a dozen hours of downtime a year, it is not clear that even the largest mainframes could handle the entire workload gracefully, with delivery rates of over 100 messages per second during peak hours.

So the new LISTSERV was going to be a distributed service, to avoid the problems we were having with BITNIC. On the other hand, the good thing with BITNIC is that it was very convenient for the users. If you wanted to subscribe to a mailing list, you just sent mail to INFO@BITNIC and the NIC staff would add you. Since the lists were all at BITNIC, there was no need to worry about where to send the request. With a distributed system, on the other hand, the lists were going to be spread over the network, and the users would have to be taught to write to LISTSERV@CEARN for the LINKFAIL list, to LISTSERV@TAMVM1 for the RSCSMODS list, and so on. Well, at the very beginning it looked like this was the price to pay to move away from the centralised model and all the problems that came with it. At any rate, users were going to have to know where the list was located anyway to post messages to the list, since there was no way we would have been able to set up mail aliases at BITNIC for all the lists. But "Revised LISTSERV" turned out to be a lot more successful than initially expected, and, with the improvements in turnaround time brought by decentralisation, more and more people started to use mailing lists. Where we had had mostly computer professionals and computer science students for users, we quickly found

ourselves dealing with increasing numbers of "real" users - the type that sends messages with "Subject: * PF03 UNDEFINED" and the like. These users had serious problems understanding the basic concepts of e-mail, mostly because, back then, electronic mail did not have the kind of attention and user support/training budget it now enjoys. Anyway, when these "real" users wanted to post a question to a list, they just replied to an old, unrelated message (which, obviously, elicited the anger of the computer types on the lists). They did not know or realise where the list was actually located. To some of them, just sending the command to the right server was a serious obstacle, and having to find out what this "right" server would be sounded more like a quest for the Holy Grail than something they wanted to bother with.

The "global service" view was an attempt at making the distributed LISTSERV system look like a centralised service to the end users. While it started with the *peered list* concept (each peer offers a "single list" image of what is in fact a distributed list), it wasn't until the inception of the automatically maintained "list of lists" in the beginning of 1988 that LISTSERV was able to offer a single-system interface to end users again. With the list of lists and the *global list* concept, users only needed to know about the location of a single LISTSERV. The system manager could set up user accounts so that people would just type TELL LISTSERV (or SEND LISTSERV, or whatever) and be connected with the closest server. The users would send commands as before (no change in syntax or other "magic option" was required), and they would be automatically forwarded to the appropriate server.

There have of course been a number of further improvements since then, most notably the *global list exchange*, which lets you post to any list without knowing where it is located. We will now see how to use all these functions in practice.

The automatically maintained "list of lists"

There are three ways to use the automatically maintained "list of lists":

- Implicitly, by issuing a command (such as SUBSCRIBE) that refers to a list maintained by a server other than the one you are sending the command to. This is entirely automatic, and there isn't much more to say about it.

- Explicitly, by issuing a LIST GLOBAL command to get a copy of the list of lists, or then to search it for topics of interest. We have already seen an example of the use of this function in the "How to..." section. If you send a LIST GLOBAL command with no parameter, LISTSERV will return a list of all the global lists (over 4100 in September 1993 - check your disk quota before ordering it!). If you specify a parameter, LISTSERV tries to match it to portions of the list name, address and title, and only returns matching entries. You will seldom get more than a couple hundred matches, so you shouldn't have to worry about disk space when requesting a search.

- Some servers also offer a more comprehensive version of this list of lists, in the form of a LISTSERV database that can be searched on any information normally available from the list headers. In particular, all the list header keywords are available as database keywords, which makes it possible to search for all the lists running with this or that option (for instance, only lists for which archives and digests are available). The database is called LISTS and servers which do not keep a copy will tell you where you can find a server with that database.

There are a few cases where the list of lists cannot provide the information you are looking for. Most of the time, it will be because the list you are looking for has been marked *confidential* or *local* by the list owner. The existence of confidential lists is hidden from non-privileged users, whereas local lists are only revealed to users of computers that have been defined as "local" (not out of secrecy, but because most sites have a number of lists for local staff, user support, and so on, and there is no point in having outside people waste their time trying to figure out what these lists are about). At any rate, the only way to join a confidential or local list is to send your request to the server that actually hosts it.

Another potential problem is that there are occasionally several global lists by the same name in the network. While LISTSERV attempts to guess which of them is most likely to be a wide-scale list, sometimes there is just not

enough information and it leaves the decision up to you. On the other hand, if you have a very specific list in mind and know where it is located, it is always best to send the command directly to that server. Fortunately, most of these conflicts have been ironed out - except of course for the ubiquitous TEST-L. Another thing you should know is that list changes are normally broadcast during the night, to reduce traffic. If you receive an announcement from the network about a new list, it may not have made it to the list of lists yet, especially if the site hosting the list is in a country where the network is often saturated.

The "Global List Exchange" (GLX)

The global list exchange (GLX) takes the global view one step further, providing a hostname through which all lists (along with their owners, and the servers that manage them) can be reached indifferently.

Using the GLX to reach mailing lists

To send mail to a LISTSERV list using the GLX, all you have to do is send your message to:

listname@LISTSERV.NET

Or, for BITNET users:

listname@LISTSERV

If there is no list by that name, you will receive an error message. BITNET users may also send non-mail files (via SENDFILE or SEND/FILE) to lists that support file delivery.

Note that the GLX is, in reality, a computer that receives your messages and relays them to their final destination. In other words, your computer does not ask the GLX where to send the message; it sends the entire message to the GLX, and lets it figure out what to do with it. If your network connection is slow and you know that the list is located on your side of the slow link, it is not a good idea to use the GLX, as the file will have to cross the link twice.

Using the GLX to order files

Perhaps the most useful function of the GLX is that it allows you to order files (including list archives) from "the server that runs the ABC list" or "the server that hosts the XYZ software collection", without having to know exactly where that server is located. For instance, if you saw on a mailing list that a program called XYZ is available from the "VM-UTIL archive", but the poster did not say where that VM-UTIL archive was located, you can ask the GLX to find it for you by sending mail to:

VM-UTIL-Server@LISTSERV.NET

with the command:

GET XYZ PACKAGE

If there is nothing matching "VM-UTIL", the command will be executed by the LISTSERV operating the GLX and you will be told that the file does not exist. Note that only filelists associated with a mailing list are reachable via the GLX; "stand-alone" filelists are assumed to be for local use and are not indexed in the GLX.

Similarly, to order a copy of the archives of the LMAIL-L list for January 1993, you would send mail to LMAIL-L-Server@LISTSERV.NET with the command GET LMAIL-L LOG9301.

Using the GLX to contact list owners

Another thing you can do with the GLX is contact list owners without having to know the location of the list. For instance, to write to the owners of the LINKFAIL list, you would send mail to:

LINKFAIL-Request@LISTSERV.NET

Note that the GLX does not forward directly to the list owners, as it does not keep that much information about the lists in its index. Instead, it forwards your request to the server hosting the LINKFAIL list, which in turn will forward it to the list owners. This also ensures that the contact addresses are always up to date. Unfortunately, there are still a few sites which do not support the *listname*-Request convention for addressing list owners. In that case, you will receive a delivery error from the target site. There is nothing the GLX can do about it, short of bombarding the system administrator with complaints until some action is taken.

Future plans for LISTSERV

The main problem with LISTSERV so far, and probably the reason why it is such a popular flaming target outside BITNET, is that it only runs on expensive IBM mainframes. This was not the result of any religious intolerance, but a simple consequence of the fact that LISTSERV was written and maintained by a single person, whose job was to run IBM mainframes rather than VAX™ clusters or unix® workstations. Quite understandably, nobody seemed to be willing to undertake a development of that magnitude without financial compensation (although hundreds of people were willing to contribute a couple hundred lines each, nobody was willing to be responsible for trying to assemble the pieces together and having to tell all the contributors that the result was just not usable). While a few organisations were willing to fund this type of development, there were always a number of unpleasant strings attached to the pot of gold, such as providing support for this but not that type of protocol in order to fulfil some political agenda or other. At any rate, none of these projects ever took off.

But now that LISTSERV has become a commercial product, all these problems are about to disappear. Development money will be available, and of course support decisions will be based on business considerations, such as expected revenue, rather than politics or religion. You just can't run a business with products nobody wants to buy, especially when there are free list managers available to anyone via anonymous FTP. In order to survive, LISTSERV will have to keep offering a lot more functionality than these free packages *and* provide support for the environments people have: TCP/IP, VMS™, unix®, *etc*.

LISTSERV on VMS™, unix® and Windows NT™

Since version 1.7f (released in April 1993), LISTSERV no longer requires NJE (BITNET) connectivity to operate in "global view" mode (LISTSERV had always been able to run in an isolated, single-server mode without NJE, but that is not very useful). As far as high-level design is concerned, the only problem that remains to be solved is the provision of rough topological data for the Internet, so that LISTSERV can continue to distribute messages efficiently should BITNET disappear in a few years. For now, LISTSERV attempts to map Internet addresses to one of the BITNET nodes belonging to the same organisation, and then uses the BITNET topological data to perform the delivery. While this works rather well, it does rely on the availability of this BITNET data. The EARN Association is now working on the provision of this kind of topological information for the Internet as well, so that, hopefully, this last dependence on BITNET can be eliminated within a year or so.

The development of the VMS™ and unix® versions of LISTSERV is expected to start 4Q93, or possibly 1Q94 if administrative delays get in the way. LISTSERV is a pretty large package, with over 70,000 lines of code, and this development will take a while. A first usable version (code name "Patchwork") is expected 4Q94. While it will not

include the file server and database functions, it is expected to have, on the average, the same or more functionality than the free list managers, with the advantage of full compatibility with the existing VM LISTSERV (same syntax, same configuration options, *etc*.) Although the first revision levels of Patchwork will probably ship with "global view" support disabled, for damage control ("test one thing at a time"), the code will be there, ready to be activated once the package is robust enough to be taken out without a leash. So within a couple months of the initial release, there should be a "Patchwork II" revision with the ability to interface with the existing LISTSERV network, participate in the GLX, and so on.

While this development goes on, the file server functions will be redesigned from scratch to put more emphasis on the hierarchical structure of the LISTSERV file system and provide a more user-friendly interface, with a syntax similar to what people use every day on their personal computer or workstation. This new file server system will be introduced simultaneously on VM, VMS™ and unix®, about 4-5 months after the release of the Patchwork stage. The new stage, called "Libris", should provide significantly more functionality than any other list manager (apart from the VM version). Libris will definitely have full support for the global view.

Development will then continue with the database functions and Windows NT™ support, and a common version is expected 1Q96, with full compatibility across the range of supported systems. At this point, it will no longer matter to the users (subscribers and list owners alike) whether you are running LISTSERV on an IBM mainframe, a workstation or a personal computer. Moving a list from the one to the other will not require anyone to learn new commands and adapt to new conventions, and no functionality will be lost. For optimal flexibility, each department will be able to maintain its own set of lists on an inexpensive personal computer and take advantage of the built-in windows-based interface for list management, while relying on a larger campus server for mail distribution. This kind of setup will make it easier for non-technical people to manage simple lists in a cost-effective manner, and should significantly increase the usefulness of LISTSERV to the academic community.

Standardisation efforts

One of the most annoying problems with LISTSERV nowadays, at least for the non-technical users, is the proliferation of "clones" which call themselves LISTSERV but use an incompatible (and sometimes vastly different) command set. To take a fictitious example, it is not uncommon to run across announcements like the following:

```
The new Chevy-lovers list is finally operational!!! Thanks to Fred
Brown, Chevy-lovers now has a new home on a REAL listserv! No more
waiting for Jack to come back from vacations to update the list (hey
Jack, I just couldn't resist!) Anyway, to join the list you have to send
mail to listserv@xyz.com with the command:

                              add chevy-lovers

And I have more good news: xyz.com is a MoonCenter 6/90 with 4 times the
speed of the old machine we ran the list on! It runs the unix version of
the bitnet listserver with some *neat* improvements by Fred Brown!
```

While it is obviously a good thing to have list managers for systems LISTSERV does not support yet, the current plethora of list managers is a significant obstacle to the successful use of mailing lists by people who lack the technical expertise to tell the would-be "real LISTSERVs" from the genuine ones. And it is usually the authors of the least comprehensive and least compatible systems who most strenuously insist that theirs is a real, genuine, absolutely authentic "listserv"... With the result that many new users got a poor impression of LISTSERV through one of the clones, and decided not to waste any more time on mailing lists.

The only way to solve this problem is to standardise the syntax of the basic list management commands, and make it clear that it is not acceptable for a list manager to call itself LISTSERV if it is not compatible with the real thing. An Internet standard on list management syntax should permit the convergence of the major list managers over the next few years, and solve the problem in all but a handful of cases. There will always be people who do not like the

standardised syntax and who will continue to use their own, but they will have to call their packages something else. Since LISTSERV went commercial, it has been a lot easier to convince people to use another name if their package is not compatible, and this part of the problem can be considered solved now.

This standardisation work will take place within the IETF framework and is expected to start in a matter of months (*i.e.* 4Q93 or 1Q94).

SENDFILE for the Internet

Another planned enhancement is the development of an efficient SENDFILE-like function for the Internet. While this is unfortunately another "hot" religious issue, the lack of SENDFILE is one of the first things users whose organisation left BITNET complain about. The reason people like SENDFILE is that it works just like a mail-order store: you place your order, the mail-order people take care of it and the package arrives in your mailbox the next day (with the speed of electronic delivery, that would be just minutes later). FTP, on the other hand, requires you to sit by the phone while the mail-order store gathers the items you have ordered, packages them, and brings them to the post office. You are not giving any instruction during this process, and in fact all you can do is ask the operator what is currently going on and how long this is going to take, and he'll say they've gathered about half of the items so far, please hold on. And, of course, sometimes they'll hang up by mistake, and you will have to call back and place the same order again (since you never seem to get to the same operator, they don't remember what your order was and you have to start from scratch every time). In the real world, this kind of organisation would drive anyone crazy in no time, and the shop in question would be out of business in a month. But in the computer world, FTP has a virtual monopoly and the question is whether you want to order or not, rather than whether you are satisfied with the service.

A new protocol is being developed by the UFT-L working group in order to offer SENDFILE functionality to people who would like to have it. LISTSERV will make use of implementations of this protocol as soon as they become available, and, in particular, the new file server functions will be designed to fully exploit its capabilities. The protocol will be public, of course, and anyone can join the UFT-L list by sending the usual SUBSCRIBE command. We do however ask participants to respect the group's stated purpose, which is to develop a quality sender-initiated file transfer protocol for TCP/IP within a year, rather than discuss the usefulness of such a protocol.

User-friendly interfaces for personal computers

Another project is the development of graphical user interfaces for personal computers, in order to make LISTSERV easier to use for people who, unfortunately, lack the time to read documents of the size of this one. There will probably be a first version based on WinSocket, for users with direct TCP/IP connectivity, and a more universal but less sophisticated version that will just compose LISTSERV commands and copy them to the clipboard, so that they can be easily pasted into any mail program. Depending on demand, a number of fully functional versions might also be developed for the major personal computer mail systems.

LMAIL and L-SOFT are trademarks of L-Soft international.
Unix is a registered trademark of UNIX Systems Laboratories, Inc.
IBM is a registered trademark of International Business Machines Corporation.
VAX, VMS and OpenVMS are trademarks of Digital Equipment Corporation.
Windows, Windows NT and NT are registered trademarks of Microsoft corporation.
All other trademarks, both marked and not marked, are the property of their respective owners.