# The ACPC Network Guide
Version 1.1

The ACPC Network Group

consisting of

H. W. Loidl, RISC-Linz (Coordinator)
H. Wabnig, University Vienna
J. Fritscher, TU-Vienna
S. Grabner, GUP-Linz
E. Cikan & P. Kutschera, University Vienna
N. Ulamec & M. Salchegger, RIST++ Salzburg

July 6, 1993

# Contents

**Abstract**

The ACPC Network Guide is a manual that gives the reader an overview of the machines in the ACPC Network and the software that is installed on these machines. Therefore, it gives the user of one of these machines the most important information that is needed for the work on this machine and it refers to additional manuals for more complex software packages where this is necessary.

However, before this overview is given, in the three introductory chapters the structure and the aims of the ACPC are discussed. The first chapters also contain some general information about the ACPC Network that might also be of interest for people who do not intend to work on one of the machines in this network. A reference to an FTP server, containing information about the ACPC Network and the ACPC in general is given. The ACPC Mailing List, which is used for distributing information among the groups in the ACPC, is also mentioned in this general part of the manual. Furthermore, this part also contains a short description of the groups in the ACPC and of their research interests in the area of parallel computation.

The main part of the manual is then formed by one section for each ACPC working group that contains some information about the group and its members, a description of the hardware at this group and a description of the software that is installed on these machines.

The appendices summarize information that is already present but spread around in the other chapters. As we think that the information presented there will be needed quite frequently, the appendices can be seen as a kind of quick reference.

# About the Authors

The authors of this manual are the members of the *ACPC Network Group*. Each working group in the ACPC has one representative, the Network Coordinator of that group, who is responsible for keeping the information about his working group in this manual up to date. The Network Coordinator is also the contact person for people who want to work on a machine of the group he is representing or just for getting more information about the machines or the software on the machines of that group.

As one important task of the ACPC Network Group is to keep this manual up to date, one member of this group has to coordinate the changes that have to be made. This coordinator of 'The ACPC Network Guide' also takes care of the general parts and of the appendices in this manual. This coordinator is currently H. W. Loidl from RISC-Linz.

Currently, the ACPC Network Group consists of the following people:

| Network Coordinator | ACPC Group | Email / Telephone |
|---|---|---|
| H. W. Loidl | RISC-Linz | hwloidl@risc.uni-linz.ac.at<br>07236 3231 66 |
| J. Fritscher | TU-Vienna,<br>Theoretical Chemistry<br>Group (Vienna) | fritscher@edvz.tuwien.ac.at<br>0222 58801 5505 |
| S. Grabner | GUP-Linz | grabner@gup.uni-linz.ac.at<br>0732 2468 884 |
| E. Cikan<br><br>P. Kutschera | University Vienna<br>(PPS)<br>University Vienna<br>(PPS) | cikan@par.univie.ac.at<br>0222 392647 201<br>peter@par.univie.ac.at<br>0222 40103 2789 |
| H. Wabnig | University Vienna | wabnig@ani.univie.ac.at<br>0222 4086366 17 |
| N. Ulamec<br><br>M. Salchegger | RIST++ Salzburg<br><br>RIST++ Salzburg | ulamec@kirk.coma.sbg.ac.at<br>0662 8044 6770<br>salchegg@spock.coma.sbg.ac.at<br>0662 8044 6771 |

# Chapter 1

# Introduction

## 1.1 What is the ACPC?

The Austrian Center for Parallel Computation (ACPC) is a joint organization formed by seven institutes at the universities of Linz, Salzburg and Vienna. The ACPC was founded in 1989 to promote research, education and industrial cooperation in the area of parallel computation in Austria and is supported by the Austrian Research Foundation and the Austrian Ministry of Science and Research. At present, approximately 50 researchers at the levels of professor, senior and junior researcher, and doctoral student from these seven groups are united in the ACPC effort.

Each group of the ACPC is specialized in a different subarea of parallel computation (parallel symbolic computation, parallel numerical applications, parallelization of programming languages, parallel computer graphics and visualization, theory of parallel computation, parallel hardware and software systems, ...). Hence, the ACPC groups profit from a lot of synergy effects by several joint research projects.

Participants in the ACPC are at present seven academic groups headed by (in alphabetical order):

Prof. Bruno Buchberger, RISC-Linz

Prof. Günter Haring, Institute for Applied Computer Science and Information Systems, University of Vienna

Dr. Wolfgang Kleinert/Doz. Christoph Überhuber, Technical University of Vienna

Prof. Peter Schuster/Doz. Othmar Steinhauser, Theoretical Chemistry Group, University of Vienna

Prof. Jens Volkert (Spokesman), Department for Computer Graphics and Parallel Processing, Institute for Computer Science, University of Linz

Prof. Hans P. Zima, Institute for Software Technology and Parallel Systems, University of Vienna

Prof. Peter Zinterhof, RIST++ Salzburg

## 1.2 Aims of the ACPC

The ACPC aims at focusing and coordinating the Austrian academic research and graduate education in the area of parallel computation. It starts from the past and present academic research activities at the major Austrian universities and organizes a series of interrelated new research projects in the area of software for parallel architectures and applications of parallel computation. The ACPC started a graduate special curriculum in parallel computation as a part of a computer science or mathematics master's or Ph.D. program. The ACPC puts a particular emphasis on motivating Austrian software and hardware companies to cooperate in pursuing the goals of the ACPC and in initiating projects that transfer foundational research in parallel computation into industrial products (software, systems).

Summarizing, the main activities for achieving these goals of the ACPC are:

- The ACPC organizes joint research projects between the various members of the ACPC with a special view to achieving synergy. Many of these projects also include partners from outside the ACPC, thereby embedding the ACPC in the international activities in the research in parallel computation.
- The ACPC has already designed a joint curriculum for diploma and doctoral students who want to specialize in the area of parallel computation so that students can easily get credit for courses taken at a different site of the ACPC.
- The ACPC cooperates with several industrial partners in joint work leading to a technological transfer and marketable products. Furthermore, the new results of ongoing projects are offered and presented to the industry such that immediate profit can be taken out of these scientific developments.
- The ACPC encourages researchers and students to visit and to study at different sites within the ACPC. By this, the researchers and the students can get profit out of the research in different subareas of parallel computation that is done at the different sites.
- To improve the knowledge transfer in the area of parallel computation not only inside the ACPC, which is done by regular internal ACPC workshops, but also with other institutes working in this area, the ACPC has organized the '1$^{st}$ International Conference of the Austrian Center for Parallel Computation' in Salzburg, in October 1991. This was the first of a series of conferences that will be held every two years. Furthermore, the ACPC regularly organizes workshops that deal with special aspects of parallel computation. Another important means

for the international knowledge transfer is the 'ACPC Technical Report Series' in which all the groups of the ACPC publish their new research results.

- The ACPC has set up a powerful network over which all partners of the ACPC can access all parallel machines available at the various sites. Therefore, all researchers in the ACPC have access to almost all modern parallel architectures. This manual aims at giving the reader an overview of these parallel architectures their characteristics, location in the network and the software that is installed on these machines.

## 1.3 The ACPC Network

The ACPC Network is a network of parallel computers located at the individual institutions that form the ACPC. In order to reach each of these computers they have been combined to the ACPC Network. Therefore, it should be possible for every person from one of the ACPC groups (but also from outside) to reach any computer in the network. Thereby, the ACPC Network grants access to essentially every interesting parallel architecture.

From the hardware side the network is based on the already existing connections that combine the universities in Austria. As the whole network is itself a part of the worldwide Internet, the machines can be reached from any part of the world that has access to the Internet. For a more detailed description of the hardware upon which the network is based see Chapter 2.

However, just building up the hardware and operating system layer for combining various machines in a network is of course not enough to give other users the opportunity of working on remote machines. For this purpose, also a description of the software and of the working environment on the machines in the network is necessary. Therefore, the main part of this manual (Chapter 4) deals with the description of the hardware and of the software that is available at each ACPC group. This should be used as a first information to get a quick overview of the available facilities in the network. However, if somebody is interested in one special software product that is installed on one of the machines he/she may refer to the documentation of the software products that is available at each of the ACPC groups[1]. However, that part of the documentation that has been distributed over all groups has been kept as small as possible. Hence, one should also contact the network coordinator of the group where the software product has been installed to get more information.

## 1.4 Guest Accounts

In the ACPC Network, on each machine some guest accounts have been installed. Hence, each ACPC group has one such guest account on each machine in the network[2].

[1] For a list of the documentation that is available at each group see Section 3.6.

[2] See Section 4 for a description of each of the machines in the ACPC Network.

The purpose of this account is to offer all members of the group the possibility to have a look on each available machine, in order to decide whether it is interesting to apply this particular machine. This opportunity is especially intended for students who want to get an overview of the available hardware without themselves having to visit all the institutes of the ACPC. As there are now many different architectures available in the ACPC Network, every person working in the ACPC can choose the most appropriate architecture for his application.

However, these guest accounts are only intended to get a very first experience. Typical actions performed under this account will be for example the reading of on-line manuals to study some details or just having a look what is new on the machine. As soon as somebody wants to do some real work this person should apply for a personal account and then work under this account. For an explanation of this application procedure see Section 1.5.

As such guest accounts are always a potential security hole for each machine, there may be some security measures taken to avoid an unauthorized usage. However, there is no global security strategy but each system administrator may decide about this independently. The user of a guest account must therefore be prepared to be confronted with one or several of the following security measures:

- The user might be asked for his email address such that the system gets information who is actually logged in.
- The permissions of some files like `.cshrc` might be set such that the user can not modify them.
- The group of the user may be set in such a way that it will not allow him any additional actions.
- The password of the guest account might have to be changed regularly.

These security measures may lead to a rather inconvenient guest account. But as this account is only intended for some simple actions like listed above this should be no problem. Hence, the general rule is that whenever some serious work is done on a machine, a personal account shall be installed.

## 1.5 The Account Application Procedure

This section will describe how to get an account on any machine in the ACPC Network. The main motivation for introducing this standard application procedure was to get the information about who is actually working on a certain machine. We have tried to make the whole procedure as convenient for the user as possible. However, some minimal effort in applying for an account is inevitable.

All you have to do for getting an account on a certain machine is to fill in the application form that is shown in Figures 1.1 and 1.2 and to send the form to the system

## Autrian Center for Parallel Computation

Account Application Form

This application is to be filled out for an account on a computer system managed by the ACPC.

System Name (as described in the ACPC Manual):

Name:

Organization:

Telephone:

E-mail address:

Mail address:

My intended research on this system is:

Project Title:

Are you a member of the ACPC?

Do you have any special requirements (hardware, software or other resources)?

Figure 1.1: Application Form (Part 1)

We must request you to comply with the following guidelines when using your account on the above mentioned computer system. If you do not, we are obliged to safeguard the interests of other users by removing your account.

Should your project require substantial machine resources, special approval must be given. Please indicate this in the description of your intended research.

You will be notified by e-mail when you account is installed.

The machine may go down for experimental work or for systems operations with little notice.

The disk is not to be used for backup purposes.

Due credit will be given in publications reporting on work done on this machine. A copy of each such publication will be sent to the system managing site (according to the ACPC Manual) unless it is published in the ACPC Technical Report series.

I agree to abide by the guidelines for use of the system as set forth above and any further rules on machine usage should they become necessary. I understand that failure to comply with any regulations will result in the removal of my account.

Name:

Signature:

Date:

Group leader:

Signature:

Figure 1.2: Application Form (Part 2)

administrator of that machine. As soon as the account has been created, the user will be informed about the login name and the password via email. The application form that is shown in Figures 1.1 and 1.2 is available at all ACPC groups and as a file at the official ACPC anonymous FTP server. The filled in form can be sent via FAX to speed up the whole process of account application.

## 1.6 The Official ACPC Anonymous FTP Server

In the ACPC the exchange of files is usually handled in such a way that the sender puts files on a special FTP server, the so called official ACPC anonymous FTP server, and announces the availability of this file to the receiver. This procedure is especially recommended if the file has to be sent to several persons or if it might be of interest to several persons or if it should be made publicly available for a longer time.

The address of the official ACPC anonymous FTP server is: `ftp.risc.uni-linz.ac.at` directory `pub/acpc`. Physically, the FTP server is located at RISC-Linz in Hagenberg.

The information in the FTP server is structured into several subdirectories:

- The directory `Manual` contains the sources of the ACPC Manual as well as the complete current version of the manual in the files `acpc-manual.dvi.Z` and `acpc-manual.ps.Z`. The latter two files are a compressed dvi and a compressed Postscript file, respectively. This directory is further structured into subdirectories:
  - For each group there is one subdirectory containing the information that is specific to this group (this includes the information in Chapter 3 and Chapter 4).
  - The directories `general` and `appendices` contain general information that appears in the first chapters of the manual and in the appendices of the manual, respectively.
  - The directory `forms` contains the current hardware and software forms that must be used to incorporate new hardware or software into the manual. Furthermore, this directory contains the account application form that is also shown in Figures 1.1 and Figures 1.2. The other files in the `forms` directory specify the structure of the overall manual and of the group specific information.
  - The directory `suggest` contains suggestions for changes in the ACPC Manual.

  However, this information should be only of interest for the ACPC Network Group, who maintains the manual. For others only the above two files, containing the manual itself will be relevant.

- The directory `Misc` contains miscellaneous information not fitting elsewhere.

- The file `News` contains important news that might be of interest for all ACPC groups.
- There are subdirectories for several ACPC groups that are maintained by these groups and that contain informations of these groups that has to be published or sent to other people.

As an alternative to using the ACPC anonymous FTP server for sending files normal email should be used if the files should be sent only once to one person. For this purpose we recommend the following procedure:

1. Make a tar file out of all files that have to be sent. This file should have the extension '.tar' (e.g. foo.tar).
2. Compress this file using standard compress. This is not only important to minimize the data that has to be actually sent but it is also important as some mailers might restrict the size of emails they are sending. Using GNU Zip instead of compress will yield even shorter files. However, if you want to use this compression format please make sure that the receiver also has the GNU Zip program available. The resulting file should have extension '.Z' if it has been compressed or '.z' if it has been zipped (e.g. foo.tar.Z).
3. As the whole email system is intended to send ASCII text files only, there will be problems with sending binary files (such as dvi files). Certain bytes in these files will be interpreted (such as CTRL-D as end of file) and therefore the files in general won't be received in the same form as they have been sent. Therefore, all binary files must be uuencoded before they can be sent (using the command uuencode <file> <name>). The resulting file should have extension '.uue' (e.g. foo.tar.Z.uue). This file now can be sent via email.

## 1.7 The ACPC Mailing List

In order to facilitate and to speed up the communication among all members of the ACPC the ACPC mailing list has been created. The idea of this mailing list is that all emails being directed to the mailing list are automatically sent to all subscribers of the mailing list. Therefore, any message can easily be sent to all interested persons in the ACPC. However, as every email that is sent to the ACPC mailing list will be received by everybody in the ACPC, this should be used only for topics that really affect all ACPC groups.

The email address of the ACPC mailing list is: `ACPC@aearn.bitnet`.

It has to be emphasized that sending to the ACPC Mailing List is not a privilege of ACPC members only. Especially, it can be used for the announcement of workshops, talks or other events in the area of parallel computation.

# Chapter 2

# Network Connections

In this chapter the network connections between the ACPC groups will be discussed. Some basic knowledge about the capacity of these hardware connections is very important to understand how the interactive work on a remote machine will look like.

First of all, the capacities of the long range connections are stated:

- Vienna – Linz[1]: *128 kb/s.*
- Linz – Hagenberg: *64 kb/s.*
- Linz – Salzburg: *64 kb/s.*
- Vienna – Salzburg: *64 kb/s.*

Apart from the direct connections between the institutes there is also a connection between Linz and Graz. This connection is important, as the logical connection between Linz and Vienna may be routed over this connection[2]. However, currently the routing between Linz and Vienna is done dynamically either over the direct connection or over Graz, depending on how much load is on each of these connections. Therefore, it can't be said beforehand which way a packet from Linz to Vienna will take. On various machines in the ACPC Network a tool called `traceroute` is installed that gives, when it is invoked with the target machine as its parameter, the way of the connection to the target machine by printing each intermediate machine or router. It is strongly recommended to use this tool (if available) if there are problems with the connection between two machines in the ACPC Network.

The capacities of the connections from and to Graz are:

- Vienna – Graz: *128 kb/s.*
- Linz – Graz: *64 kb/s.*

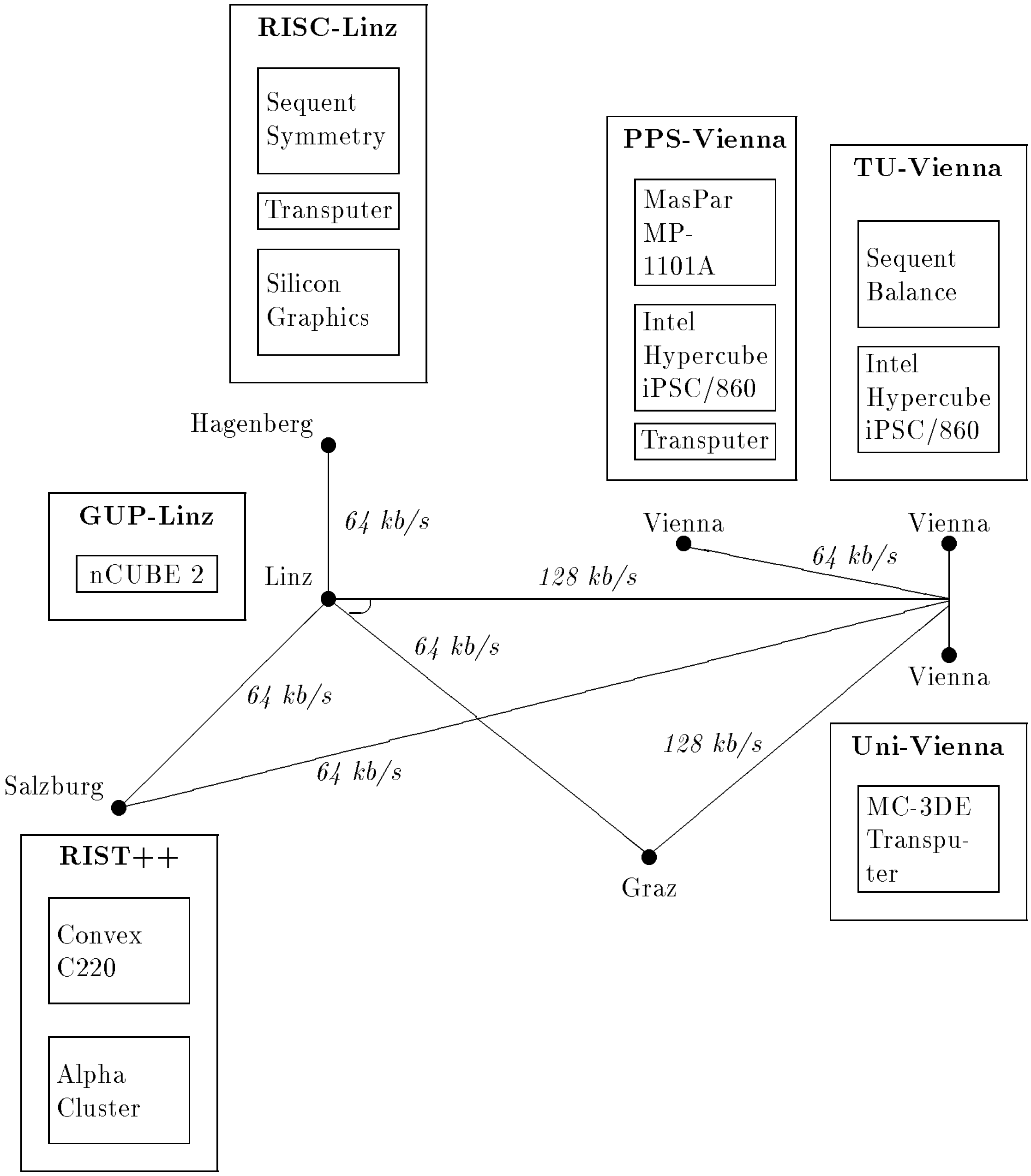


Figure 2.1: 'Map' of the ACPC Network

| | | *RISC* |
|---|---|---|
| RISC-Linz | Sequent Symmetry | 5/6/17 (0 %) |
| | Transputer | 7/10/12 (0 %) |
| Uni-Vienna | MC-3/DE Transputer | 125/306/767 (0 %) |
| TU-Vienna | Sequent Balance | 113/728/4027 (0 %) |
| | Intel Hypercube | 102/482/2864 (0 %) |
| GUP-Linz | nCube2 | 64/67/69 (0 %) |
| Uni-Vienna | MasPar | 143/554/3400 (0 %) |
| (PPS) | Intel Hypercube | 150/692/4340 (0 %) |
| | Transputer | 99/342/858 (0 %) |
| RIST++ Salzburg | Convex | — |
| | Alpha Cluster | 107/143/389 (0 %) |

Table 2.1: Ping Times of the ACPC Machines from RISC-Linz

To give the reader a global overview of the whole ACPC Network, Figure 2.1 shows a 'map' of this network containing all machines currently available. The quarter arc between the lines to Vienna and to Graz originating in Linz denotes the dynamic routing over the connection with the smallest load. Furthermore, the capacities of the long range connections are also shown in this figure. For a more detailed information on the various computers in the network see Chapter 4.

However, for an exact characterization of the speed of a connection the information about its theoretical capacity is not sufficient. Better information is provided by the `ping` command that allows the user to check the time required for sending a packet of a certain size to another machine. Some tests with this command have been performed on most of the machines in the ACPC Network. The results of these tests can be seen in Figures 2.1 to 2.5.

However, these times must not be taken as absolute values for the general quality of a certain connection! It must be emphasized that *these times may vary very much according to the daytime* when it is tested. But these timings should give the reader a rough feeling how fast a certain connection would be.

### Experience Reports

The rest of this chapter contains some experience reports about the work on remote machines.

[1] This connection will be upgraded to *256 kb/s* at the end of May 1993.

[2] Note that this also affects the machines in Hagenberg and probably those in Salzburg when they set up a connection to a machine in Vienna.

| | | *Uni-Vienna* |
|---|---|---|
| RISC-Linz | Sequent Symmetry | 124/320/700 (0 %) |
| | Transputer | 266/961/1444 (0 %) |
| Uni-Vienna | MC-3/DE Transputer | local |
| TU-Vienna | Sequent Balance | 37/54/161 (0 %) |
| | Intel Hypercube | 34/53/231 (0 %) |
| GUP-Linz | nCube2 | 332/588/1210 (0 %) |
| Uni-Vienna | MasPar | 77/77/80 (0 %) |
| (PPS) | Intel Hypercube | 80/81/98 (0 %) |
| | Transputer | 30/30/35 (0 %) |
| RIST++ Salzburg | Convex | — |
| | Alpha Cluster | 121/791/4979 (43 %) |

Table 2.2: Ping Times of the ACPC Machines from the University of Vienna

| | | *TU-Vienna* |
|---|---|---|
| RISC-Linz | Sequent Symmetry | 100/407/1421 |
| | Transputer | — |
| Uni-Vienna | MC-3/DE Transputer | 29/30/42 |
| TU-Vienna | Sequent Balance | 7/10/14 |
| | Intel Hypercube | 3/3/5 |
| GUP-Linz | nCube2 | 50/1258/5364 |
| Uni-Vienna | MasPar | 48/58/139 |
| (PPS) | Intel Hypercube | — |
| RIST++ Salzburg | Convex | — |
| | Alpha Cluster | — |

Table 2.3: Ping Times of the ACPC Machines from the TU-Vienna

| | | *GUP-Linz* |
|---|---|---|
| RISC-Linz | Sequent Symmetry | 62/112/685 |
| | Transputer | 61/84/206 |
| Uni-Vienna | MC-3/DE Transputer | 83/952/6389 (7 % loss) |
| TU-Vienna | Sequent Balance | 109/305/1956 |
| | Intel Hypercube | 60/199/619 |
| GUP-Linz | nCube2 | 1/1/2 |
| Uni-Vienna (PPS) | MasPar | 121/504/1234 |
| | Intel Hypercube | 116/600/1843 |
| | Transputer | 55/125/251 |
| RIST++ Salzburg | Convex | — |
| | Alpha Cluster | — |

Table 2.4: Ping Times of the ACPC Machines from GUP-Linz

| | | *Uni-Salzburg* |
|---|---|---|
| RISC-Linz | Sequent Symmetry | 103/158/150 (0 %) |
| | Transputer | — |
| Uni-Vienna | MC-3/DE Transputer | — |
| TU-Vienna | Sequent Balance | 101/322/691 (0 %) |
| | Intel Hypercube | 84/269/609 (0 %) |
| GUP-Linz | nCube2 | 43/121/366 (0 %) |
| Uni-Vienna (PPS) | MasPar | — |
| | Intel Hypercube | 120/462/1019 (0 %) |
| RIST++ Salzburg | Convex | 0/0/0 (0 %) |
| | Alpha Cluster | 0/0/0 (0 %) |

Table 2.5: Ping Times of the ACPC Machines from the University of Salzburg

### Connection RISC – University Vienna

The interactive work on this connection is quite slow. At least it is significantly slower than on the connection between RISC and the TU-Vienna. Hence, if some serious work should be done over this connection it is recommended to do the editing of files locally and then to transfer the files to the remote computer. This style of work may be a little inconvenient but as soon as one gets accustomed to this way of working there should be no bigger problems with it.

### Connection RISC – TU-Vienna

This connection is faster than the connection to the University Vienna. This can also be seen from Table 2.3. With the speed on this connection it is very well possible to do a remote editing of files. Although, there will be a little delay this should not hinder the work.

### Connection RISC – University Linz

This connection is the fastest one from RISC within the ACPC Network. If one works from RISC remotely on a machine at the University Linz there is hardly any delay recognizable. Therefore, the work could be done as if one would work locally.

# Chapter 3

# The Groups in the ACPC

This chapter contains a short introduction of each of the ACPC groups. By this the reader should get some information on what parts of parallel computation each of the groups is mainly interested in.

This chapter should only give a first impression of the areas in which the different groups are working. For further information (e.g. about ongoing projects) the interested reader should contact the several groups directly.

## 3.1 RISC-Linz — Research Institute for Symbolic Computation, Johannes Kepler University

At RISC-Linz, whose chair is Prof. Bruno Buchberger and which is located in the castle of Hagenberg near Linz, Parallel Computation is handled in close connection with Symbolic Computation. Symbolic Computation is that part of computer science respectively mathematics that is dedicated to the solution of problems with non-numerical ("symbolic") objects. It serves as the basis for many advanced software topics, such as

- computer aided design and geometric modelling,
- robotics and softautomation,
- computer algebra systems,
- expert systems and artificial intelligence,
- logic and functional programming,
- scientific software.

In all these areas, there exist many very interesting and powerful, but also time-consuming, algorithms. Thus, many problems become intractable very soon when handled on conventional sequential computer architectures. Therefore, Parallel Computation has become an integrated part of Symbolic Computation.

The research in Parallel Computation at RISC-Linz has its roots in 1978, when a parallel architecture and programming language for Symbolic Computation were created. Today, the Parallel Computation Group at RISC-Linz consists of two faculty members, four PhD students and several diploma students. The main research interests of this group are:

- the design and implementation of *parallel algorithms* for computer algebra, computational logic and computer geometry,
- the design and implementation of *parallel software systems* for implementing these algorithms, and
- the *parallelization of* declarative (functional and logic) *programming languages*.

A special emphasis is put on the synergy between these research areas in Parallel Computation and also between the work in other areas of Symbolic Computation and the work in Parallel Computation.

## 3.2 Institute for Applied Computer Science and Information Systems, University of Vienna

The main research topics in the parallel processing field are:

- performance evaluation of multiprocessor systems
- process to processor mapping and process scheduling
- development environments and methodology for parallel software
- parallel and distributed simulation
- neural network models and their simulation on parallel hardware

The term performance evaluation covers all aspects of using multiprocessor hardware to gain efficiency. A special focus is on performance modelling based on timed Petri nets (deterministic and stochastic). Such models are used to characterize the factors influencing the performance of working parallel systems (e.g. algorithms for parallel programs, hardware descriptions, process to processor mapping etc.) and are solved either analytically or by simulation. Performance indices derived by the model evaluation are used to analyze the efficiency of parallel programs, the utilization of resources (hardware) and last but not least provide the basis for process to processor mappings. Process to processor mapping is a key problem in parallel processing which cannot be treated analytically in its instance as a graph matching problem because of NP-completeness. A lot of assumptions in current approaches to solve the problem turn out to be very weak, whereas a lot of characteristics of both the parallel program (like synchronization) and the parallel hardware are not represented in the mapping models. Simulated annealing as a general technique in combinatorial optimization offers a promising facility to the practical treatment of (small) practical problem instances.

Parallel programming techniques (divide-and-conquer, compute-aggregate-broadcast, pipelining, etc.) should be provided to the developer in a rather automated manner. They should be a part of a general parallel program development environment, containing tools for (advanced) editing, compilation, debugging, parallelization and code optimization, performance prediction and monitoring, visualization etc. Usability and efficiency of such tools depend highly on the user interface and the modes of interaction. Both these aspects, tool-support and sophisticated user interfaces, are becoming more and more important with the upcoming multiprocessor workstation technology. The availability of a performance oriented parallel program development methodology is quite attractive from a scientific as well as from an economic point of view.

Today the neural processing approach is successfully operating in constraint satisfaction, pattern association, content-addressable memory, control, motion, vision, coordination applications etc. Special purpose neural networks have shown the power of the non-algorithmic processing methodology encountering qualities like distributed knowledge representation, tendential behaviour, learning, generalization, fault tolerance etc., and beating traditional AI methodologies in several cases. The massive parallelism offered by neural network simulation models can naturally often be exploited, and parallel execution can speed up simulations to bring them into practical use. The usage of parallel processing hardware promises not only the quantitative improvement of reducing simulation times, but gives also a perspective for a qualitative improvement of neural network models.

## 3.3 GUP-Linz — Department for Computer Graphics and Parallel Processing, Institute of Computer Science, Johannes Kepler University Linz

At GUP-Linz (whose chair is Prof. Jens Volkert), research in parallel computation began in late 1990. This department consists of Prof. Volkert, three Ph.D. students and several diploma students. The main research interests of this group are:

- Monitoring & Debugging of Parallel Programs (especially for distributed memory machines)
- Visualization of Data Partitioning and Graphical Domain Decomposition for Distributed (or Virtual Shared) Memory Machines

- Visualization of Multidimensional Data (in parallel)
- Hardware Modelling for Performance Analysis of Parallel Programs

The main basis for this work is a nCUBE 2 distributed memory hypercube with location at the Computing Center of the University Linz.

## 3.4 Institute for Software Technology and Parallel Systems, University of Vienna (PPS — Programming Languages, Parallel Systems)

One of the most fundamental challenges facing computer science today is the need to develop the algorithms and programming tools required to harness the vast computing power of modern high-performance computers. The commercially available machines that fall into this (roughly defined) class today include vector and array computers, parallel architectures with a large number of processors, and networks of high-performance workstations. These machines provide their users with a peak performance of up to 10 billion floating point operations per second, thus opening the way for the solution of problems which seemed computationally quite out of reach until recently, and spawning interest in a whole range of new problems.

High-performance computers have become a fundamental tool of science. While computers have long been used to organize and control scientific experiments and analyze their results, the new machines can do much more than that: they play an important role in theoretical science too, enabling scientists to construct and test models in their search for theories to describe natural phenomena.

Our interest focusses on programming environments for high-performance computers, in particular on the following areas:

- automatic parallelization
- performance analysis
- advanced compiling techniques
- use of knowledge-based methods for the intelligent support of program transformation strategies
- intelligent man-machine interfaces

We believe that these areas are of vital importance if high-performance computers are to be effectively utilized.

## 3.5 RIST++ — Research Institute for Software Technology + Paris Lodron University Salzburg

Chair of the RIST++ ist Prof. Peter Zinterhof. The RIST++ is located at the University of Salzburg and at the Technology Center of Salzburg.

The RIST++ was founded in 1986 as 'Forschungsinstitut für Softwaretechnologie'. Currently the RIST++ consists of the following labs:

- Parallel Numeric and Supercomputing-Lab
- Basic Research-Lab
- Image-Lab
- Fuzzy-Lab
- Signals and Systems-Lab
- Applied Number Theory-Lab
- Neural Computation-Lab

In each lab problems in basic research and applied research for implementation to sequential and parallel computers are the topics. Some of the labs are partly financed by the industry. The equipment used consists of some transputer systems (now mainly for minor projects), the CONVEX 220 situated at the Technology Center Salzburg, and a Cluster consisting of 7 DEC 3000 Alpha AXP 400 workstations.

## 3.6 List of Documentation that is available at each ACPC Group

The following sections show for each group the list of documentation that has been copied and sent to all other ACPC groups. So, you can see here the documentation that is available at all ACPC groups. For more detailed documentation you have to contact the network coordinator of the group where the software product is installed.

### 3.6.1 RISC-Linz

- Guide to Parallel Programming on Sequent Computers
- Programming Guide (ch. 1, 2, 4, index)
- C & Language Tools
  - C Users Manual

  - pdbx Users Manual

- Reference Manual
  - About this Manual
  - Section 3 (PPS)
  - Permuted Index
- $\mu$-System Annotated Reference Manual
- STRAND88Release Notes for Users (Sequent Symmetry)
- STRAND88Technical Description – Admirality Release (Aug.89)
- C-Linda Reference Manual

### 3.6.2 Institute for Applied Computer Science and Information Systems, University of Vienna

- "PARIX Release 1.1. Software Documentation", Parsytec Computer GmbH, D-5100 Aachen, Juelicher Strasse 338, Germany, 1992.
- "PARIX Release 1.1. Manual Pages", Parsytec Computer GmbH, D-5100 Aachen, Juelicher Strasse 338, Germany, 1992.

### 3.6.3 TU-Vienna

- iPSC/860 Users Guide
- iPSC/860 Network Queueing Systems Manual
- iPSC/860 Debugger Manual
- iPSC/860 Programmers Reference Manual
- iPSC/860 Concurrent Programming Vol. 3
- iPSC/860 Simulators Manual
- iPSC/860 Technical Documentation Guide
- iPSC/860 Fortran Compiler Users Guide
- iPSC/860 C Compiler Users Guide

### 3.6.4 GUP-Linz

- "nCUBE 2 Programmer's Guide", nCUBE Corporation, 1990.
- "nCUBE 2 Programmer's Reference Manual", nCUBE Corporation, 1990.
- "nCUBE 2 Programmer's Reference Manual", nCUBE Corporation, 1990.
- "nCUBE 2 Math Library Manual: Introduction and SSS32", nCUBE Corporation, 1990.
- "nCUBE 2 Math Library Manual: SSS64 and BLAS", nCUBE Corporation, 1990.

# Chapter 4

# Facilities in the Network

This chapter contains a group description for each of the ACPC groups. This contains a description of the hardware that is available at each of these groups and of the software that is installed at each of these computers. The whole description is divided in several sections, one for each ACPC group. Note that Appendix A contains a table of the machines that are described in more detail in this chapter and that Appendix B contains a list of all persons in the ACPC. These appendices summarize information that is given in a different form in this chapter.

## 4.1 Group Description of RISC-Linz

### General Information about the Group

- **Group Name:** Parallel Computation Group at the Research Institute for Symbolic Computation (RISC-Linz).
- **Group Leader:** Prof. B. Buchberger.
- **Address:**

  RISC-Linz
  Research Institute for Symbolic Computation
  Johannes Kepler University
  A-4040 Linz, Austria, Europe
- **Phone Number:** 07236-3231 (secretary: 07236-3231-21).
- **FAX Number:** 07236-3231-30
- **Domain Name:** Internet: `risc.uni-linz.ac.at`. Bitnet: `ALIJKU11`.
- **Persons:**

| Name | Internet-account | Bitnet-account | Phone Number |
|---|---|---|---|
| Bruno Buchberger | buchberg | K313370 | 41 |
| Hoon Hong | hhong | K314577 | 48 |
| Tudor Jebelean | tjebelea | — | 64 |
| Hans Wolfgang Loidl | hwloidl | K318576 | 66 |
| Wolfgang Schreiner | schreine | K315640 | 66 |
| Kurt Siegl | siegl | K316340 | 62 |
| Dan Teodosiu | danteo | | |
| Peter Zettler | zettler | K312970 | 67 |
| Werner Landerl | landerl | — | 66 |

### 4.1.1 Sequent Symmetry S81

#### Footprint

| | |
|---|---|
| System | Sequent Symmetry S81 |
| Architecture | Shared Memory MIMD |
| # of Processors | 20 |
| Type of Processors | Intel 80386/80387 |
| Memory Size | 128 MB |
| Disk Capacity | 4.32 GB |
| Operating System | DYNIX/ptx 2.0 (based on UNIX System V R 3.2) |
| Name | melmac.risc.uni-linz.ac.at |
| Internet Address | 193.170.36.100 |
| Location | RISC |
| System Administrator | K. Erdei (`kerdei`) |
| Network Coordinator | H. W. Loidl (`hwloidl`) |

#### Availability

| Group | RISC | TU-Vienna | Vienna | Linz | Salzburg | World |
|---|---|---|---|---|---|---|
| | good | good | slow | good | good | good |

As the 64Kb/s connection between Linz and Hagenberg is available now, the connection to the machine from outside RISC should be quite good.

#### Further Description

This machine is not only used for research purpose but also as the main computing server for the institute. Therefore, computational intensive jobs should be run during the night hours or at weekends.

### Software

### $\mu$System

- **Name:** $\mu$System 4.4.2
- **Location:** /usr/local/usystem
- **Type:** Parallel Programming Package.
- **Characterization:** Library of C routines (and C structures) for handling light-weight tasks. Designed for Shared-Memory Machines. Furthermore: Synchronization via Semaphores and Message Passing. Non-blocking I/O is supported. Exception handling is implemented.
- **Functions:**
  - Invocation: `/usr/local/bin/concc`, `/usr/local/bin/parcc`.
  - Termination:
  - Help: `man concc`, `man parcc`.
  - Example: (Path has been set to `/usr/local/bin`). Compile program for single-process execution:
    `concc /usr/local/usystem/examples/MessagePassing/ProdCons.c`
    Compile program for multi-process execution:
    `parcc /usr/local/usystem/examples/MessagePassing/ProdCons.c`
  - Other important functions: For monitoring of programs use the following compiler: `/usr/local/bin/moncc`
- **Installation:**
- **Manuals:** Manual pages for `concc`, `parcc` and `moncc`. "$\mu$System Annotated Reference Manual" (also available as '.ps' file).

### aXe

- **Name:** aXe 3.1
- **Location:** /usr/local/bin/axe (/usr/local/lib/X11/aXe)
- **Type:** ASCII-Text editor
- **Characterization:** aXe is a text editor for program source code or LaTeX input or simple text files. It is NOT a WYSIWYG editor. aXe features multiple windows, multiple buffers per window, a file selection box, a so called "mini-buffer" which allows to call external commands. It allows cut and paste, and has many functions selectable from menus.

- **Functions:**
  - Invocation: `axe`
  - Termination: There's a Quit button.
  - Help: There's a help menu with the sub-topics General Help, Key Bindings, Customization, Information about Popups.
  - Example:
  - Other important functions: It is possible to go to a line with a specific number.
- **Installation:** Runs only under X Windows.
- **Manuals:** `man axe`, but this mostly refers to the Help menu.

**Bash**

- **Name:** GNU Bourne Again SHell 1.12
- **Location:** /usr/local/gnu/bash-1.12
- **Type:** A bourne shell compatible command language interpreter
- **Characterization:** Extension of the traditional bourne shell that also includes most of the features of the c-shell and of the korn-shell. Furthermore, this shell allows command line editing with vi-like commands. Implementation of IEEE Posix Shell and Tools specification.
- **Functions:**
  - Invocation: `/usr/local/bin/bash`
  - Termination: `exit`
  - Help: `help`
  - Example: `bash -rcfile ~/.cshrc`
    Start the bash and use the file '~/.cshrc' as startup file (Note: As c-shell and bash scripts are not fully compatible some errors will probably occur).
  - Other important functions:
- **Installation:** Put the setting of your environment in the file '~/.bashrc'.
- **Manuals:** man pages; a paper describing the features of bash can be found in '/usr/local/gnu/bash-1.12/documentation'.

### Bison

- **Name:** GNU Bison 1.14
- **Location:** /usr/local/gnu/bison-1.14
- **Type:** Parser generator
- **Characterization:** yacc replacement
- **Functions:**
  - Invocation: `/usr/local/bin/bison`
  - Termination:
  - Help: `man bison`
  - Example:
  - Other important functions:
- **Installation:**
- **Manuals:** Bison Reference Manual (bison.texinfo), man pages

### Crisp

- **Name:** Crisp 1.9
- **Location:** /usr/local/crisp
- **Type:** Editor
- **Characterization:** DOS Brief look alike Editor
- **Functions:**
  - Invocation: `/usr/local/bin/cr`
  - Termination: `ALT-x`
  - Help: `ALT-h`
  - Example:
  - Other important functions: Multiple windows, programming language support, mailing, copy, cut and paste of columns
- **Installation:** Set environment vars by: `source ~crisp/crisp.setenv`
- **Manuals:** Interactive help system

### DYNIX/ptx

- **Name:** Dynix/ptx 2.0
- **Location:** melmac.risc.uni-linz.ac.at
- **Type:** UNIX System V Release 3.2.0 i386
- **Characterization:** Standard System V operating system with parallel extensions for a shared memory parallel computer.
- **Functions:**
  - Invocation: `rlogin melmac`
  - Termination: `exit`
  - Help: `help`
  - Example:
  - Other important functions:
- **Installation:** Ask for an acount by email to `schreine`.
- **Manuals:** UNIX System On-Line Help, man pages,
  DYNIX/ptx reference manual, DYNIX/ptx user guide

### Emacs

- **Name:** GNU Emacs 18.59
- **Location:** /usr/local/gnu/emacs
- **Type:** Editor
- **Characterization:** Fully featured editor
- **Functions:**
  - Invocation: `/usr/local/bin/emacs`
  - Termination: `CTRL-x CTRL-c`
  - Help: `CTRL-h`
  - Example:
  - Other important functions: Multiple windows, programming language support, mailing
- **Installation:**
- **Manuals:** Interactive Help system, man pages

### Flex

- **Name:** GNU Flex 2.3 26
- **Location:** /usr/local/gnu/flex-2.3.7
- **Type:** Lexical analyzer generator
- **Characterization:** Tool for generating scanners: programs which recognize lexical patterns in text.
- **Functions:**
  - Invocation: `/usr/local/bin/flex`
  - Termination:
  - Help: `man flex`
  - Example:
  - Other important functions:
- **Installation:**
- **Manuals:** flex, flexdoc man pages

### GNU Assembler

- **Name:** gas 1.38
- **Location:** /usr/local/gnu/gas-1.38
- **Type:** i386 Assembler
- **Characterization:** Downward compatible to unix "as".
- **Functions:**
  - Invocation: `/usr/local/bin/a386`
  - Termination:
  - Help: `man as`
  - Example:
  - Other important functions:
- **Installation:**
- **Manuals:** "as" manual descripton

### GNU C

- **Name:** gcc 2.3.3
- **Location:** /usr/local/gnu/gcc-2.3.3
- **Type:** C-Compiler
- **Characterization:** Fast Ansi C Compiler
- **Functions:**
  - Invocation: `/usr/local/bin/gcc`
  - Termination:
  - Help: `man gcc`
  - Example:
  - Other important functions:
- **Installation:**
- **Manuals:** gcc man page

### GNU Locate

- **Name:** glocate
- **Location:** /usr/local/bin/glocate
- **Type:** Fast file find tool
- **Characterization:** Searches a database for pathnames that match the pattern that has been given on the command line. Currently, the database contains all files on melmac except those in temporary directories like /tmp and /usr/tmp. The database is rebuilt every day at 4 am.
- **Functions:**
  - Invocation: `/usr/local/bin/glocate`
  - Termination:
  - Help: `man glocate`
  - Example: `glocate *.tar.Z`
    Lists all compressed tar files.
  - Other important functions:
- **Installation:**
- **Manuals:**

### GNU Zip

- **Name:** gzip 0.8.1
- **Location:** /usr/local/gnu/gzip-0.8.1
- **Type:** File compression and expansion
- **Characterization:** The GNU Zip tools `gzip, gunzip, zcat` compress and decompress files by using Lempel-Ziv coding (LZ77). The usage of `gzip` and `gunzip` as quite similar to `compress` and `uncompress`. Compressed files get the extension `.z`. This compression method is better than the standard `compress`. `gzip` can be used as filter if no file is specified.
- **Functions:**
  - Invocation: `/usr/local/bin/gzip`
  - Termination:
  - Help: `man gzip, /usr/local/bin/gzip -h`
  - Example: `gzip -v -9 foo`
    Compresses the file `foo` yielding a file `foo.z`. For that the best compression method is used (`-9`) and the compression factor is printed due to verbose mode (`-v`).

    `gunzip foo.z` or `gzip -d foo.z`
    Decompress the file that has been compressed above.
  - Other important functions: Multiple compressed files can be concatenated. **Note:** There are also various other (de-)compression tools available like: `zip` (DOS port), `zoo`, `squeeze0` and `unarj`. See the corresponding man pages for further information.
- **Installation:**
- **Manuals:**

### Gopher Client

- **Name:** gopher, xgopher version II release 5
- **Location:** /usr/local/unixutil/gopher+1.2b2
- **Type:** ASCII and X11 client for accessing gopher servers.
- **Characterization:** These clients allow the communication with arbitrary gopher servers. The Internet Gopher is a distributed document delivery service. The documents are organized in hierarchical way. The Internet Gopher Servers accept simple requests from the clients and respond by sending the client a document.

- **Functions:**
  - Invocation: `/usr/local/bin/gopher`, `/usr/local/bin/xgopher`
  - Termination: `Quit Button`
  - Help: `Help Button`, `man gopher`, `man xgopher`
  - Example: `xgopher melmac.risc.uni-linz.ac.at`
    This will connect the user with the gopher server at RISC-Linz. The user can then walk through the hierarchy of available files in the same way as in a usual directory hierarchy. Documents can also be directly fetched from the server.
  - Other important functions:
- **Installation:** For `gopher`: `setenv PAGER /usr/local/bin/less`. Several other environment variables might be set (e.g. for defining with which program mail should be sent). See the man page of `gopher`. One can create an own `~/.gopherrc` file containing bookmarks, configuration information etc.
- **Manuals:**

**Less Paginator**

- **Name:** Less
- **Location:** /usr/local/less
- **Type:** File displaying tool.
- **Characterization:** Similar to “more”, but allows backwards movement.
- **Functions:**
  - Invocation: `/usr/local/bin/less`
  - Termination: `q`
  - Help: `h`
  - Example:
  - Other important functions:
- **Installation:**
- **Manuals:** less man page, online help

**Maple**

- **Name:** MAPLE V
- **Location:** /usr/local/maple
- **Type:** Interactive Computer Algebra System
- **Characterization:** Maple has the ability to algebraically manipulate unbounded integers, exact rational numbers, real numbers with arbitrary precision, symbolic formulae, polynomials, sets, lists, and equa- tions. It can solve systems of equations and differentiate and integrate formulae. **Note**: There is also a Maple version with an X11 interface: `xmaple`.
- **Functions:**
  - Invocation: `/usr/local/bin/maple`
  - Termination: `quit`
  - Help: `?`
  - Example:
  - Other important functions:
- **Installation:**
- **Manuals:** Maple Reference Manual, Interactive Help, man pages

||MAPLE||

- **Name:** ||MAPLE|| Shell 1.0
- **Location:** `/usr/local/maple/pmaple`
- **Type:** Interactive shell of ||MAPLE|| (Parallel Maple), Parallel Computer Algebra System
- **Characterization:** Computer Algebra System, in which logic programming provides parallelism and imperative programming provides efficiency. The system is built as a combination between the parallel declarative programming language Strand and the sequential computer algebra system Maple.

  The system supports the whole Maple library and all Strand features.
- **Functions:**
  - Invocation: `/usr/local/bin/parmaple`
  - Termination: `<CTRL> C`
  - Help: `man parmaple`
  - Example:

- Other important functions:

- **Installation:**

- **Manuals:** man pages,
  /usr/local/maple/man/pmapledoc.dvi - the documentation

**Netrek**

- **Name:** XtrekII 6.1
- **Location:** /usr/local/netrek
- **Type:** Game
- **Characterization:** Multi-users X-windows Star Trek game
- **Functions:**
  - Invocation: `/usr/local/bin/netrek`
  - Termination: `Q`
  - Help: `h`
  - Example:
  - Other important functions:
- **Installation:** A private environment can be set by creating an own `~/.netrekrc` file.
- **Manuals:** Online start-up instructions

**PACLIB**

- **Name:** PACLIB (Version 1.18)
- **Location:** `/usr/local/paclib`
- **Type:** Parallel System for Computer Algebra.
- **Characterization:** PACLIB consists of a compiler for a parallel programming language (based on C) and a library of sequential and parallel computer algebra algorithms (based on SACLIB). The user can implement parallel algorithms and monitor their dynamic behavior using the visualization tools `pacgraph` and `pacutil`.
- **Functions:** The PACLIB compiler is invoked as `pacc` (no optimization) or `paco` (optimization) or `pacd` (debugging) with the same syntax as the GNU C compiler. The final program can be called with several runtime options e.g.

```
pacc test.c -o prog
prog +p16 +N1000000 +t1
```

The option `+p` defines the number of processors, `+N` defines the number of heap words, and `+t1` triggers the generation of profiling information that is written into a file `pac.dat`. This information can be analyzed by the tools `pacgraph` and `pacutil`. The commands

```
pacgraph 250 250 <pac.dat >grap.tex
pacutil  250 250 <pac.dat >util.tex
```

generate two LaTeX pictures of size 250 times 250 point that display the activity of the tasks and the utilization of the processors, respectively.

- **Manuals:** PACLIB User Manual (RISC-Linz Technical Report 92-32 and ACPC Technical Report ACPC/TR 92-9).

### Patch

- **Name:** GNU Patch 2.0.12u4
- **Location:** /usr/local/gnu/patch-2.0.12u4
- **Type:** Program patching tool
- **Characterization:** Applies a diff file to an original.
- **Functions:**
  - Invocation: `/usr/local/bin/patch` *orig patchfile*
  - Termination:
  - Help: `man patch`
  - Example:
  - Other important functions:
- **Installation:**
- **Manuals:** patch man page

### Perl

- **Name:** GNU Perl 4.019
- **Location:** /usr/local/gnu/perl-4.019
- **Type:** Practical Extraction and Report Language

- **Characterization:** An interpreted language optimized for text processing like scanning text files and extracting information. It combines features of C, sed, awk and sh.
- **Functions:**
  - Invocation: `/usr/local/bin/perl`
  - Termination: `<CTRL> C`
  - Help: `man perl`
  - Example:
  - Other important functions:
- **Installation:**
- **Manuals:** Perl Reference Guide, Frequently Asked Questions about Perl (both in '/usr/local/gnu/perl-4.019').

### Raytracer

- **Name:** Rayshade V3
- **Location:** /usr/local/ray3
- **Type:** Ray tracing program
- **Characterization:** Rayshade reads a file describing a scene to be rendered and produces a Utah Raster RLE format file of the ray-traced image.
- **Functions:**
  - Invocation: `/usr/local/bin/rayshade`
  - Termination:
  - Help: `man rayshade`
  - Example:
  - Other important functions:
- **Installation:**
- **Manuals:** rayshade man page

### SACLIB

- **Name:** SACLIB 1.1, ISAC 2.0
- **Location:**
  - `/usr/local/saclib`: The maintenance directory.

  - `/usr/local/saclib/saclib1.1`: The root of the directory tree for the current version.

- **Type:** Computer Algebra Library for the C programming language.
- **Characterization:** SACLIB contains programs for list processing, infinite precision arithmetic (integer, rational and modular), operations on multivariate polynomials (arithmetic operations, substitution, greatest common divisors, resultants, factorization, and others), polynomial real root isolation and refinement, and operations with real algebraic numbers and polynomials having algebraic number coefficients.

  There is also a simple interactive shell called `isac` which can be used for experimenting with the functions in the library.

- **Functions:**

  These are only the functions of `isac`, the interactive shell. In general, the library functions are intended to be called from a program.

  - Invocation: `isac`
  - Termination: `quit;`
  - Help: `help;`
  - Example:

```
1> I := 50;

2>  F := IFACTL(I);

3> IWRITE(F);
30414093201713378043612608166064768844377641568960512000000000000
4> quit;
```

  - Other important functions: The shell scripts `sman` and `sdesc` can be used for getting help on SACLIB functions. See the manual for a description of these commands.

- **Installation:** Your environment must have the variable `saclib` set to `/usr/local/saclib/saclib1.1` and your path must contain `/usr/local/saclib/saclib1.1/bin`.
- **Manuals:** `/usr/local/saclib/saclib1.1/doc/saclib.dvi` and `/usr/local/saclib/saclib1.1/doc/saclocal.dvi`.

### STRAND88

- **Name:** STRAND88 Buckingham Release

- **Location:** /usr/local/strand
- **Type:** Parallel Declarative Programming Language
- **Characterization:** STRAND$\underline{88}$ is an implementation of the Strand language, which includes an environment for developing and running Strand programs. The Strand language is desinged to utilize the facilities of concurrent machines and parallel architectures. It supports concurrent processing, stream communication between processes, Data-flow, modular development, declarative reading, object oriented programming and interfaces to other languages and data types.
- **Functions:**
  - Invocation: st
  - Termination: <CTRL>C
  - Help: st -h
  - Example:
  - Other important functions:
- **Installation:** source /usr/local/strand/strandrc
- **Manuals:** STRAND$\underline{88}$ User Manual,
  Book: Strand New Concepts in Parallel Programming

**TeX, LaTeX**

- **Name:** TeX C Version 3.14t3
- **Location:** /usr/local/tex
- **Type:** Text formatting and typesetting system
- **Characterization:** Standard Scientific Word processor
- **Functions:**
  - Invocation: /usr/local/bin/tex, /usr/local/bin/latex
  - Termination:
  - Help: man tex, man latex
  - Example:
  - Other important functions:
- **Installation:** Set environment vars by: source ~tex/lib/TeX3.setenv
- **Manuals:** LaTex - A Document Preparation System by Leslie Lamport
  The TeXbook by Donald E. Knuth, man pages

### TEXCAD

- **Name:** TeXcad 1.2
- **Location:** `/usr/local/tex/src/xtexcad`
- **Type:** Simple interactive tool for the generation of LaTeX pictures.
- **Characterization:** The program allows you to create in an interactive way LaTeX pictures consisting of straight lines, arrows, boxes, circles and text. Copy and erase operations are supported.
- **Functions:**
  - Invocation: `texcad`
  - Termination: Click on button FILE and menu item TERMINATE.
  - Example: `texcad test.pic`
- **Manuals:** `/usr/local/tex/src/xtexcad/texcad12.man` (LaTeX file).

### Tgif

- **Name:** Tgif Version 2.12-p7
- **Location:** /usr/local/lib/tgif
- **Type:** X-lib based 2-D drawing facility under X11.
- **Characterization:** Tgif is an interactive drawing tool that allows the user to draw and manipulate objects in the X Windows System (it is based on Xlib). Tgif supports a bottom-up construction of hierarchical objects as well as a top-down specification of drawings. Among the formats that can be incorporated into a drawing are X11 bitmaps, Encapsulated Postscript and many others. The supported output formats are Postscript, Encapsulated Postscript, X11 bitmap and X11 pixmap.
- **Functions:**
  - Invocation: `/usr/local/bin/tgif`
  - Termination: `<CTRL> Q`
  - Help: `man tgif`
  - Example: `tgif foo.ps`
    Allows the user to modify the postscript file `foo.ps`.
  - Other important functions:
- **Installation:** The following environment variables should be set:
  - `setenv TGIFPATH /usr/local/lib/tgif`,
  - `setenv TGIFICON tgificon.obj`.
- **Manuals:**

### Utah Raster Toolkit

- **Name:** Utah Raster Toolkit 3.0
- **Location:** /usr/local/urt
- **Type:** C graphic library
- **Characterization:** The Utah Raster Toolkit is a collection of programs and C routines for dealing with raster images commonly encountered in computer graphics. A device and system independent image format stores images and information about them.
- **Functions:**
  - Invocation: include functions to your C-programs
  - Termination:
  - Help: `man urt`
  - Example:
  - Other important functions:
- **Installation:**
- **Manuals:** urt man pages

### X-Windows

- **Name:** ptx/windows X11 R4, R5
- **Location:** /usr/bin/X11, /usr/local/X11R5
- **Type:** Portable, network-distributed window system
- **Characterization:** A window system developed at MIT. X uses a client-server architecture to promote portable applications.
- **Functions:**
  - Invocation:
  - Termination:
  - Help: `xman`
  - Example:
  - Other important functions: OSF-Motiv Window manager (mwm)
- **Installation:**
- **Manuals:** ptx/windows Programmers Reference,
  ptx/windows Users Reference, man pages,
  The X Windows System

### Xmandel

- **Name:** xmandel, pxmandel
- **Location:** /usr/local/x11/xmandel
- **Type:** Demonstration program.
- **Characterization:** X11 interface for generating and colouring Mandelbrot and Julia sets. There is a sequential program (xmandel) as well as a parallel program (pxmandel) available, that can be used for demonstrating the speed-up achievable by parallel computation.
- **Functions:**
  - Invocation: `xmandel`, `pxmandel <no. of processors>`
  - Termination: Quit Button
  - Help: `man xmandel`
  - Example: `pxmandel 16 -geometry 400x450+450+0 &`
    Creates a window that is a little smaller than half of the monitor size. The parallel computation is started by clicking at the 'Mandelbrot' or 'Julia' button.
    `pardemo/pxdemo`
    Starts two windows: One for sequential computation and one for parallel computation on 16 processors. This shell script is highly recommended for demos!
    **Note:** Don't move the cursor inside the pictures while they are computed!
  - Other important functions: By clicking at the zoom button and then at a part of the picture, this part will be zoomed and recomputed.
- **Installation:** Don't forget to set your DISPLAY environment variable before starting the program.
- **Manuals:**

### XParty

- **Name:** XParty (XPhone)
- **Location:** /usr/local/x11/xparty
- **Type:** Communication Tool
- **Characterization:** Multi-user/multi-display communication utility. Allows several people to perform a "Network Conference". The user who starts the conference leads the discussion and has several privileges. Motif based. Can also be used for one-way communication.

- **Functions:**
  - Invocation: `/usr/local/bin/xparty`
  - Termination: `Quit`-Button
  - Help: `help`-Button
  - Example: Start program with `xparty`. Press `AddR`-Button to add a recipient. Select the new recipient by pressing the corresponding button in the 'Recipients' part of the screen. If the recipient accepts the call he will be added to the 'on the air' list and the communication can begin.
  - Other important functions: One way communication by pressing the `one-way` button before sending.
- **Installation:** Parameters can be set in your own '~/.Xdefaults' file. See the file'/usr/lib/x11/app-defaults/XParty' for the default settings. The hidden file '~/.xmsg_hosts' contains a list of recipients (in the format 'label@display'). When a new recipient is added in xparty, a new entry is automatically created in this file.
- **Manuals:** xparty man page

### 4.1.2 Transputer Network

#### Footprint

| System | Impuls Transputer Network |
|---|---|
| Architecture | Distributed Memory MIMD |
| # of Processors | 16 |
| Type of Processors | T800 |
| Memory Size | 16 * 4 MB = 64 MB |
| Disk Capacity | 0 MB |
| Operating System | Helios (optionally !) |
| Name | — |
| Internet Address | — |
| Location | RISC |
| System Administrator | K. Siegl (`siegl`) |
| Network Coordinator | H. W. Loidl (`hwloidl`) |

#### Availability

| Group | RISC | TU-Vienna | Vienna | Linz | Salzburg | World |
|---|---|---|---|---|---|---|
| | good | good | slow | good | good | good |

### Further Description

The transputer network is not always turned on. Contact the network administrator if you want to work on it. This transputer network has no fixed topology. Before starting the actual application the user has to set up a special topology.

### Footprint of the Host Computer

| | |
|---|---|
| System | AT-PC |
| Type of Processors | Intel 80386 |
| Memory Size | 8 MB |
| Disk Capacity | 100 MB |
| Operating System | Interactive System V R 3.2 |
| Name | venus.risc.uni-linz.ac.at |
| Internet Address | 193.170.36.70 |
| Location | RISC |
| System Administrator | W. Schreiner (`schreine`) |
| Network Coordinator | H. W. Loidl (`hwloidl`) |

### Software

### ANSI C Toolset

- **Name:** ANSI C Toolset of INMOS (Version IMS D711)
- **Location:** /usr0/dos/ictools/tools
- **Type:** C compiler, linker and configurer
- **Characterization:** Like the occam toolset this toolset contains several programs for programming in ANSI C on a transputer. Among others, this toolset contains the following programs:
  - `icc` — the ANSI C compiler,
  - `ilink` — the toolset linker,
  - `icconf` — the configurer tool,
  - `icollect` — the code collector.
- **Functions:** For the most important tools listed above, their usage is described in more detail here:
  - `icc` — the ANSI C compiler. An ANSI C Compiler with concurrency support of INMOS that has been developed from the 3L Parallel C Compiler, which implemented K & R C.
    * Invocation: `icc` *filename* { *options* }

    * Termination: `<CTRL> C`
    * Help:
    * Example: `icc foo.c`
      Compiles the C program in `foo.c` and creates the result in `foo.tco`.
    * Other important functions:
  - `ilink` — the toolset linker. The toolset linker for linking separately compiled program units and libraries into a single code unit.
    * Invocation: `ilink` { *filenames* } { *options* }
    * Termination: `<CTRL> C`
    * Help:
    * Example: `ilink foo1.tco foo2.tco`
      Links the compiled C programs `foo1.tco` and `foo2.tco` into the file `foo1.lku`.
    * Other important functions:
  - `icconf` — the configurer tool. The configurer tool generates configuration information for transputer networks from a configuration description written in the transputer configuration language.
    * Invocation: `icconf` *filename* { *options* }
    * Termination: `<CTRL> C`
    * Help:
    * Example: `icconf foo.cfs`
      Produce the configuration data file `foo.cfb` from the configuration description file `foo.cfs`. The `foo.cfb` file can then be taken as the input file for `icollect`.
    * Other important functions:
  - `icollect` — the code collector. The code collector generates executable files for single and multitransputer programs, from configuration data files and linked units respectively.
    * Invocation: `icollect` *filename* { *options* }
    * Termination: `<CTRL> C`
    * Help:
    * Example: `icollect foo.lku /t`
      This creates a bootable file for a single tranpsuter in `foo.btl`.
      `icollect foo.cfb`
      (This creates a file `foo.btl` that can be executed on a multitransputer network).
    * Other important functions:

- **Installation:** `SET IBOARDSIZE=#400000` (this sets the transputer memory size for the bootstrap loader).

  `SET TRANSPUTER=#150` (this sets the transputer base address).

`SET ITERM=Z:\USR0\DOS\ITOOLS\ITERMS\IBMPC.ITM` (the terminal description file for a PC).

`SET ISEARCH=Z:\USR0\DOS\ITOOLS\LIBS` (this sets the library path).

- **Manuals:** "ANSI C Toolset — User Manual", by INMOS Limited, 1990.

### Helios

- **Name:** Helios 1.2.1
- **Location:** /usr0/dos/helios
- **Type:** Parallel Operating System
- **Characterization:** Helios is an Unix look alike Operating system for distributed memory machines (i.e. Transputers)
- **Functions:**
  - Invocation: `/usr0/dos/helios/helios0` - Single transputer
    `/usr0/dos/helios/hyper` - (Hypercube) Network version.
  - Termination: `<CTRL><SHIFT>F9` or
    Login under user: shutdown
  - Help: `help`
  - Example:
  - Other important functions:
- **Installation:** Start up VPIX (DOS emulation) first.
  `SET IBOARDSIZE=#400000`
- **Manuals:** Helios User Manual, man pages

### LSC Transputer Toolset

- **Name:** Logical Systems Transputer Toolset (Version 89.1.c)
- **Location:** /usr0/lsc
- **Type:** C preprocessor, compiler, assembler, linker.
- **Characterization:** Like the ANSI C toolset this toolset contains several programs for programming in C on a transputer. Among others, this toolset contains the following programs:
  - `pp` — the C preprocessor,
  - `tcx` — the C compiler,

  - `tasm` — the assembler,
  - `tlnk` — the linker,
  - `tlib` — the relocatable file librarian,
  - `make` — a program maintenance utility,
  - `ld-one` — the single transputer loader,
  - `ld-net` — the network loader for arrays of transputers,
  - `cio` — the host I/O driver for C I/O on the transputer,
  - `tio` — a simple example driver for standard I/O.

- **Functions:** For the most important tools listed above, their usage is described in more detail here:
  - `pp` — the C preprocessor. This C preprocessor conforms to the ANSI Draft Proposed "C" standard and contains options for backward compatibility with programs written for non-ANSI preprocessors.
    * Invocation: `pp` *filename* { *options* }
    * Termination: `<CTRL> C`
    * Help:
    * Example: `pp foo.c -o foo`
      The preprocessor takes `foo.c` as input and creates the file `foo` as output. The default suffix for the produced file is '`.pp`'.
    * Other important functions:
  - `tcx` — the C compiler. Cross compiler for the INMOS 16 and 32 bit Transputers. It does not include preprocessing.
    * Invocation: `tcx` *filename* { *options* }
    * Termination: `<CTRL> C`
    * Help:
    * Example: `tcx foo -v -p8`
      Compile the program `foo.pp` in verbose mode (option `-v`) for a T800 Transputer (option `-p8`). The created file is `foo.tal`.
    * Other important functions:
  - `tasm` — the transputer assembler. A relocating assembler for INMOS Transputers. It supports standard INMOS mnemonics and allows splitting a program into separate pieces, which are combined at linkage time.
    * Invocation: `tasm` *filename* [*temp-directory*] { *options* }
    * Termination: `<CTRL> C`
    * Help:
    * Example: `tasm foo /fast`
      Assemble the program `foo.tal` and use the directory `fast` for storing the temporary files. The created file is `foo.trl`.
    * Other important functions:

  - tlnk — the transputer linker. This linker has been designed for use with INMOS Transputers. It supports multiple libraries and allows complete control of load address for each program or library code fragment.
    * Invocation: tlnk [ *command-file* ]
    * Termination: <CTRL> C
    * Help:
    * Example: tlnk
      By this call the linker will ask for the files to be linked, for some flags, load address, stack address and other parameters.
    * Other important functions:
  - ld-one — the single transputer loader. This tool is used to load an executable file onto a single transputer. Executable files are created as a result of linking using tlnk (default suffix '.tld'). The first argument is the file to be downloaded. The second file runs on the host and handles transputer I/O requests.
    * Invocation: ld-one *filename filename*
    * Termination: <CTRL> C
    * Help:
    * Example: ld-one foo tio
      This starts foo.tld on a single transputer and tio on the host for I/O handling.
    * Other important functions:
  - ld-net — the network transputer loader. This tool is used to load a collection of executable files onto a network of transputers. Executable files are created as a result of linking using tlnk (default suffix .tld). The first argument is the file specifying the file to read to determine network topology and the list of programs to be loaded. The remaining parameters on the command line are passed to the host driver program.
    * Invocation: ld-net *network-info-file arguments*
    * Termination: <CTRL> C
    * Help:
    * Example: ld-net load
      Take the network-information-file load.nif and load the program, that is specified in load, on a network, whose topology is also specified in load. The program will be executed with the program cio.exe handling the I/O.
    * Other important functions:

- **Installation:** SET TMP=Z:\USR\TMP (directory for temporary files).

  SET PPINC=Z:\USR0\LSC\TINCLUDE (directory for include files).

  SET TLIB=Z:\USR0\LSC\TLIBRARY (directory for libraries).

  Add the following directory to your path: Z:\USR0\LSC\DOS.

- **Manuals:** “Logical Systems Transputer Toolset”, by Logical Systems, 1989.

**The Network Configurer**

- **Name:** Transputer Network Configuration Program.
- **Location:** /usr0/dos/tutil
- **Type:** Utility for fixing a given topology.
- **Characterization:** This is an occam program (so it must be started with the `iserver`) for fixing the topology of the transputer network that is described in the topology file with which this program is called. Topology descriptions for a network with up to 16 transputers can be found in /usr0/dos/tutil/topo.
- **Functions:**
  - Invocation: `/usr0/dos/itools/tools/iserver /sb /usr0/dos/tutil/conf.b8h` *topology*
  - Termination: `<CTRL> C`
  - Help:
  - Example: `iserver /sb /usr0/dos/tutil/conf.b8h /usr0/dos/tutil/topo/torus.top`

    This fixes the topology of a torus of 16 transputers. /usr0/dos/itools/tools must be in the path.
  - Other important functions:
- **Installation:** /usr0/dos/itools/tools should be put into the path.
- **Manuals:**

**Occam 2 Toolset**

- **Name:** Occam 2 Toolset for IBM/NEC PC by Inmos (Version D705B).
- **Location:** /usr0/dos/itools/tools (of host PC)
- **Type:** A collection of independent programs for programming in occam.
- **Characterization:** This toolset consists of various programs that allow the user to work with a transputer (network). The toolset contains the following programs:
  - `iboot` — the bootstrap tool,
  - `icheck` — the occam 2 syntax checker,
  - `iconf` — the configurer,

- `idebug` — the debugger,
- `idump` — the memory dumper,
- `ilibr` — the librarian,
- `ilink` — the linker,
- `ilist` — the binary lister,
- `imakef` — the makefile generator,
- `iserver` — the host file server,
- `isim` — the T414 simulator,
- `iskip` — the skip loader tool,
- `occam` — the occam 2 compiler.

- **Functions:**

  For the most important tools listed above, their usage is described in more detail here:

  - `iboot` — the bootstrap tool. This tool produces executable code for single transputers. The object file it generates can be loaded onto a single transputer and run. The input must be a compiled (`.t`*xx*) or a linked (`.c`*xx*) object file.
    * Invocation: `iboot` *filename* { *options* }
    * Termination: <CTRL> C
    * Help:
    * Example: `iboot foo.c8h`
      This takes the compiled and linked file `foo.c8h` and creates the file `foo.b8h` that can be started with the `iserver` on one transputer.
    * Other important functions:
  - `icheck` — the occam 2 syntax checker. This tool performs a syntax check of the full occam 2 product language, as described in the "Occam 2 Refernece Manual", but produces no object code. It generates more data and displays more information about the errors than the compiler.
    * Invocation: `icheck` *filename* { *options* }
    * Termination: <CTRL> C
    * Help:
    * Example: `icheck foo.occ`
      This checks the syntax of the occam program in `foo.occ`. The '`.occ`' suffix may be omitted.
    * Other important functions:
  - `iconf` — the configurer. This tool allocates occam processes to processors and assigns channels to links on transputer networks. It takes a configuration description and produces either an object code file ready for booting into a network of transputers, or a configuration map describing the allocation of code and placement of channels.

  * Invocation: `iconf` *filename* { *options* }
  * Termination: <CTRL> C
  * Help:
  * Example: `iconf foo.pgm`
    This configures a transputer network that is described in the file `foo.pgm`. The '`.pgm`' suffix may be omitted.
  * Other important functions:

- `idebug` — the debugger. This tool allows to debug an occam program even when it is executed on a transputer network. With this debugger errors can be located in the source code, parameters can be examined channels and processes can be analysed and procedures can be traced.
  * Invocation: `idebug` *bootablefile* { *options* }
  * Termination: <CTRL> C
  * Help:
  * Example: `idebug foo.btl`
    This starts the debugger on the program `foo.btl` that runs on a transputer network. Possible suffices are '`.b`*xx*', '`.btl`' and '`.btr`'.
  * Other important functions:
- `idump` — the memory dumper. This tool dumps the contents of the root processor's memory to disk. It is used to enable the debugging of code running on the root transputer.
  * Invocation: `idump` *filename memorysize* { *startoffset length* }
  * Termination: <CTRL> C
  * Help:
  * Example: `idump foo.btl #200000 #300000 #100000`
    This dumps the first half of the memory of the root transputer (bytes 0 – #200000). Additionally, #100000 bytes starting from address #300000 are dumped.
  * Other important functions:
- `ilink` — the linker. This tool creates single object files from separately compiled code and libraries (all external references are resolved).
  * Invocation: `ilink` { *filenames* } { *options* }
  * Termination: <CTRL> C
  * Help:
  * Example: `ilink foo1.t8h foo2.t8h string.lib`
    This links the compiled occam files `foo1.t8h` and `foo2.t8h` with the routines from the string library in `string.lib`. The resulting file has the name `foo1.c8h`.
  * Other important functions:
- `iserver` — the host file server. This tool loads programs onto transputers and transputer networks and provides the run-time environment through which programs communicate with the host.

    * Invocation: `iserver` { *options* }
    * Termination: <`CTRL`> `C`
    * Help:
    * Example: `iserver /sb foo.btl`
      This loads the program foo.btl on a tranputer network and boots the program (`/sb` option). To get a file like `foo.btl` the source files have to be compiled and linked and made bootable by using the bootstrap tool `iboot` (for single transputer programs) or the configurer `iconf` (for multitransputer programs). Furthermore, before the `iserver` can be started the topology of the network has to be fixed with a call to the network configurer (see below).
    * Other important functions:
  - `occam` — the occam 2 compiler. This tool compiles occam source code programs into the assembler code of the transputer (optionally to T212, T222, M212, T414, T425, T800).
    * Invocation: `occam` *filename* { *options* }
    * Termination: <`CTRL`> `C`
    * Help:
    * Example: `occam /t8 /h foo.occ`
      This compiles the occam program `foo.occ` into the assembler code of the T800 transputer in halt mode (i.e. the transputer halts when an error occurs in the program). The resulting file is `foo.t8h`.
    * Other important functions:

- **Installation:** `SET IBOARDSIZE=#400000` (this sets the transputer memory size for the bootstrap loader).

  `SET TRANSPUTER=#150` (this sets the transputer base address).

  `SET ITERM=Z:\USR0\DOS\ITOOLS\ITERMS\IBMPC.ITM` (the terminal description file for a PC).

  `SET ISEARCH=Z:\USR0\DOS\ITOOLS\LIBS` (this sets the library path).

- **Manuals:** "Occam 2 Toolset, User Manual, IBM/NEC PC", by Inmos Limited, 1989.

  "Occam 2 Reference Manual", by Inmos Limited, Prentice Hall, 1988.

### ParaScope Debugger

- **Name:** ParaScope Transputer Network Debugger for Logical Systems C.
- **Location:** /usr0/lsc/dos/cdbug
- **Type:** Debugger for parallel programs.

- **Characterization:** This debugger works with Logical Systems C on a network of Transputers. It is a 'postmortem' debugger. In several windows the source code, the memory contents and other information is shown.
- **Functions:**
  - Invocation: cdbug { *options* }
  - Termination: <CTRL> C for terminating the program that has to be debugged and to enter the debugger with the above command.

    Go with the mouse to the *Process Register Window*. Press center mouse button and choose "exit" from the menu that pops up.
  - Help:
  - Example: cdbug

    Debug a program that has just crashed on the transputer (network) or that has been stopped with <CTRL> C. In the program that has to be debugged the function debug() has to be called at the beginning of main and for each that will be linked into the main program a symbol table must be generated by calling the program symtab.exe.
  - Other important functions: Pressing the center mouse button in different windows causes different menus to pop up.
- **Installation:** SET CDPATH=Z:\USR0\LSC\DOS\CDBUG (directory for the debugger).
- **Manuals:** "ParaScope Transputer Network Debugger for Logical Systems C", by Computer Systems Architects (CSA), 1990.

### STRAND$^{\underline{88}}$ for Helios

- **Name:** STRAND$^{\underline{88}}$ Buckingham Release
- **Location:** /usr0/dos/strand
- **Type:** Parallel Declarative Programming Language
- **Characterization:** STRAND$^{\underline{88}}$ is an implementation of the Strand language, which includes an environment for developing and running Strand programs. The Strand language is designed to utilize the facilities of concurrent machines and parallel architectures. It supports concurrent processing, stream communication between processes, Data-flow, modular development, declarative reading, object oriented programming and interfaces to other languages and data types.
- **Functions:**
  - Invocation: st
  - Termination: <CTRL>C

  - Help: `st -h`
  - Example:
  - Other important functions:

- **Installation:** Start up the Helios environment first.
  `source /helios/users/strand/strandrc`
- **Manuals:** STRAND88 User Manual,
  Book: Strand New Concepts in Parallel Programming

### 4.1.3 Silicon Graphics Iris VGX Graphics Supercomputer

#### Footprint

| | |
|---|---|
| System | SGI Iris Power Series |
| Architecture | Shared Memory MIMD |
| # of Processors | 4 |
| Type of Processors | MIPS R3000 |
| Memory Size | 64 MB |
| Disk Capacity | 2 GB |
| Operating System | IRIX System V.3 |
| Name | edelweiss.risc.uni-linz.ac.at |
| Internet Address | 193.170.33.151 |
| Location | RISC |
| System Administrator | K. Erdei (`kerdei`) |
| Network Coordinator | H. W. Loidl (`hwloidl`) |

#### Availability

| Group | RISC | TU-Vienna | Vienna | Linz | Salzburg | World |
|---|---|---|---|---|---|---|
| | good | — | — | — | — | — |

#### Further Description

The installation of this machine has not been completed, yet. Therefore, there are currently no ordinary user accounts on the machine and there are also no guest accounts for the other ACPC groups installed. As soon as this machine will be opened for general usage, the guest accounts will be installed. This will be announced over the ACPC mailing list.

#### Software

## 4.2 Group Description of the Institute for Applied Computer Science and Information Systems, University of Vienna

### General Information about the Group

- **Group Name:** ACPC Group at the Institute for Applied Computer Science and Information Systems
- **Group Leader:** Prof. Dr. Günter Haring
- **Address:**

  University of Vienna
  Inst. for Applied Computer Science and Information Systems
  Lenaugasse 2/8
  A-1080 Vienna, Austria, Europe

- **Phone Number:** ++431 4086366...-10
- **FAX Number:** ++431 4080450
- **Domain Name:** Internet: `ani.univie.ac.at`. Bitnet: `awiuni11.bitnet`.
- **Persons:**

| Name | Internet-account | Bitnet-account | Phone Number |
|---|---|---|---|
| Günter Haring | haring | a4424dad | +12 |
| Alois Ferscha | ferscha | a4424dab | +18 |
| Gabriele Kotsis | kotsis | a4424dau | +14 |
| Georg Pigel | pigel | a4424gak | +13 |
| Harald Wabnig | wabnig | a4424gal | +17 |
| Ursula Presslauer | presslauer | — | +23 |

### 4.2.1 MultiCluster-3DE

#### Footprint

| | |
|---|---|
| System | MC-3DE (Local Development Station) |
| Architecture | Distributed Memory MIMD |
| # of Processors | 16 |
| Type of Processors | T800 |
| Memory Size | 16×4 MB = 64 MB |
| Disk Capacity | 0 MB |
| Operating System | Parix Release 1.1 |
| Name | — |
| Internet Address | — |
| Location | Lenaugasse 2/8 1080 Vienna |
| System Administrator | H. Wabnig (wabnig) |
| Network Coordinator | H. Wabnig (wabnig) |

#### Availability

| Group | RISC | TU-Vienna | Vienna | Linz | Salzburg | World |
|---|---|---|---|---|---|---|
| | slow | good | good | | slow | good |

#### Further Description

Our transputer system is a local development station. Although the parix concept contains software for multiuser administration it is not applicable to the MC-3DE hardware. Therefore these software parts are not included and multiuser processing is not possible. Nevertheless the system can be used over the network. But the system administrator has to be contacted to set up the system accordingly and prevent problems caused by multiple users.

The MC-3DE has a fixed topology (mesh). Special topology configurations can be built with cables.

#### Footprint of the Host Computer

| | |
|---|---|
| System | SUN IPX |
| Type of Processors | Sparc |
| Memory Size | 32 MByte |
| Disk Capacity | 1.2 GByte |
| Operating System | SunOS Release 4.1.1-IPX |
| Name | epikur.ani.univie.ac.at |
| Internet Address | 131.130.32.113 |
| Location | Lenaugasse 2/8 1080 Vienna |
| System Administrator | U. Presslauer (ursula) |
| Network Coordinator | U. Presslauer (ursula) |

**Software**

**PARIX**

- **Name:** PARIX Release 1.1
- **Location:** /home/parix
- **Type:** Operating System, C-Compiler
- **Characterization:** PARIX is a new operating system for transputer based multiprocessor systems. It consists of the basic software to load and run programs, software for administration (not included) and ACE EXPERT compilers.
- **Functions:**
  - Example:
    Change to the directory: /home/parix/examples.
    Copy the file hello.c to your home directory.
    To compile and run the hello.c example type:

    ```
    ancc.px -o foo.px foo.c
    run -f0 1 1 foo.px
    ```

    There are other examples in the examples directory. For each example exists a information file "doc" which describes how to compile and run the example program.
  - Termination: <CTRL> C
  - Help: Manual pages are available for all PARIX commands.
- **Installation:**

  For each account a .cshrc file is provided which contains all commands that are necessary to set up the environment.

- **Manuals:**

  "PARIX Release 1.1. Software Documentation", Parsytec Computer GmbH, D-5100 Aachen, Juelicher Strasse 338, Germany, 1992.

  "PARIX Release 1.1. Manual Pages", Parsytec Computer GmbH, D-5100 Aachen, Juelicher Strasse 338, Germany, 1992.

## 4.3 Group Description of TU-Vienna

### General Information about the Group

- **Group Name:** ACPC Group at the Technical University of Vienna
- **Group Leaders:** Dr. W. Kleinert and Univ. Doz. Dr. C. Überhuber
- **Address:**

  Computing Center
  Technical University of Vienna
  Wiedner Hauptstrasse 8-10
  A-1040 Vienna, Austria, Europe

- **Phone Number:** ++431-58801 (secretary: ++431-58801-5481)
- **FAX Number:** ++431-58-74-211
- **Domain Name:** Internet: `edvz.tuwien.ac.at`. Bitnet: `awituw64.bitnet`
- **Persons:**

| Name | Internet-account | Bitnet-account | Phone Number |
|---|---|---|---|
| Wolfgang Kleinert | kleinert | kleinert | 5480 |
| Irmgard Husinsky | husinsky | husinsky | 5484 |
| Josef Fritscher | fritscher | fritscher | 5505 |
| Christoph Überhuber | – | – | 5412 |
| Arnold Krommer | – | – | – |

### 4.3.1 Sequent Balance B21

#### Footprint

| | |
|---|---|
| System | Sequent Balance B21 |
| Architecture | Shared Memory MIMD |
| # of Processors | 28 |
| Type of Processors | NS32032 |
| Memory Size | 24 MB |
| Disk Capacity | 540 MB |
| Operating System | DYNIX 3.0.14 |
| Name | eacpc1.tuwien.ac.at |
| Internet Address | 128.130.36.3 |
| Location | TU-Vienna |
| System Administrator | J. Fritscher |
| Network Coordinator | J. Fritscher |

### Availability

| Group | RISC | TU-Vienna | Vienna | Linz | Salzburg | World |
|---|---|---|---|---|---|---|
| | | good | good | | | good |

### Further Description

This machine is used for research and educational purpose.

### Software

### KAP

- **Name:** Kuck & Assoc KAP/Sequent 7.2
- **Location:** /usr/bin/kap
- **Type:** Parallelizing FORTRAN compiler
- **Characterization:** Fully automatically parallelizing FORTRAN compiler
- **Functions:**
  - Invocation: `kap` or `fortran -K`
  - Termination:
  - Help: `man kap` or `man fortran`
  - Example: `fortran -K program.f`
  - Other important functions: KAP converts `IF .. GOTO` loops into `DO .. loops`
- **Installation:**
- **Manuals:** KAP/Sequent User's Guide

### C-Linda

- **Name:** SCA C-Linda 2.4
- **Location:** /usr/local/src/linda/v2.4
- **Type:** Linda pre-compiler with TupleScope debugger
- **Characterization:** A C compiler using the Linda parallel programming paradigm
- **Functions:**

  - Invocation: `clc`
  - Termination:
  - Help: `man clc`
  - Example: `clc -linda ts 2000 sample.cl`
  - Other important functions:

- **Installation:**
- **Manuals:** C-Linda Reference Manual

**GNU C**

- **Name:** gcc 2.3.3
- **Location:** /usr/local/bin/gcc
- **Type:** C-Compiler
- **Characterization:** Fast Ansi C Compiler
- **Functions:**
  - Invocation: `gcc`
  - Termination:
  - Help: `man gcc`
  - Example:
  - Other important functions:
- **Installation:**
- **Manuals:** gcc man page

### 4.3.2 Intel Hypercube iPSC/860

#### Footprint

| | |
|---|---|
| System | Intel Hypercube iPSC/860 |
| Architecture | Distributed Memory MIMD |
| # of Processors | 16 |
| Type of Processors | Intel i860 |
| Memory Size | 72 MB |
| Disk Capacity | 1.55 GB |
| Operating System | NX Rel 3.3.1 |
| Name | eacpc3.tuwien.ac.at |
| Internet Address | 128.130.36.5 |
| Location | TU-Vienna |
| System Administrator | J. Fritscher |
| Network Coordinator | J. Fritscher |

#### Availability

| Group | RISC | TU-Vienna | Vienna | Linz | Salzburg | World |
|---|---|---|---|---|---|---|
| | | good | good | | | good |

#### Further Description

#### Footprint of the Host Computer

| | |
|---|---|
| System | Intel PC 386 |
| Type of Processors | i386 |
| Memory Size | 8 MB |
| Disk Capacity | 380 MB |
| Operating System | AT&T Unix System V Rel 3.2 |
| Name | eacpc2.tuwien.ac.at |
| Internet Address | 128.130.36.4 |
| Location | TU-Vienna |
| System Administrator | J. Fritscher |
| Network Coordinator | J. Fritscher |

#### Software

#### C-Linda

- **Name:** SCA iPSC/Hypercube Linda 2.4

- **Location:** /usr/local/linda
- **Type:** Linda pre-compiler with TupleScope debugger
- **Characterization:** A C compiler using the Linda parallel programming paradigm
- **Functions:**
  - Invocation: `clc`
  - Termination:
  - Help: `man clc`
  - Example: `clc -o sample -linda ts 2000 sample.cl`
    `host -t 4 sample`
  - Other important functions: A Linda code development system (CDS) running on Sun SparcStations is included with the package. CDS is an Linda system emulator and can be used to develop Linda programs without having a parallel computer.
- **Installation:** It is necessary to include /usr/local/linda/bin to the path environment variable.
- **Manuals:** C-Linda Reference Manual

**AT&T C++**

- **Name:** CC 2.1
- **Location:** /usr/local/bin
- **Type:** C++ Compiler
- **Characterization:** AT&T's original C++ Compiler
- **Functions:**
  - Invocation: `CC`
  - Termination:
  - Help:
  - Example:
  - Other important functions:
- **Installation:**
- **Manuals:**

### PICL (Portable Intrumented Communication Library)

- **Name:** PICL Version 1/1/92 from netlib
- **Location:** /usr/local/lib
- **Type:** Library for parallel programming
- **Characterization:** In combination with PARAGRAPH an programming and analysis tool is provided. PARAGRAPH can run on any other workstation with an X-Window environment. The libraries can be called from FORTRAN or C.
- **Functions:**
  - Example: `cc -o hostprogram hostprogram.c hostlib.a -host`
    `icc -o nodeprogram nodeprogram.c nodelib.a -node`
- **Installation:**
- **Manuals:**

## 4.4 Group Description of the Theoretical Chemistry Group

### General Information about the Group

- **Group Name:** Theoretical Biochemistry Group
- **Group Leader:** Prof. O. Schuster
- **Address:**

  Institute for Theoretical Chemistry
  University of Vienna
  Währingerstraße 17
  A-1090 Vienna

- **Phone Number:** +43 222 43 61 41 ext. 78
- **FAX Number:** +43 222 40 28 525
- **Domain Name:** `itc.univie.ac.at, AWIUNI11.BITNET`
- **Persons:**

| Name | Internet-account | Bitnet-account | Phone Number |
|---|---|---|---|
| P. Schuster | pks@tiger.itc.univie.ac.at | A8441DAM | 43 61 41 ext. 78 |
| I. Hofacker | ivo@otter.itc.univie.ac.at | — | 43 61 41 ext. 78 |

## 4.5 Group Description of GUP-Linz

### General Information about the Group

- **Group Name:** ACPC Group at the University Linz
- **Group Leader:** Prof. Dr. J. Volkert
- **Address:**

  Institute of Computer Science
  Dept. for Graphics and Parallel Processing
  University of Linz
  Altenbergerstr. 69
  A-4040 Linz, Austria, Europe

- **Phone Number:** ++43-732-2468-887 (secretary)
- **FAX Number:** ++43-732-2468-10
- **Domain Name:** Internet: `gup.uni-linz.ac.at`
- **Persons:**

| Name | Internet-account | Bitnet-account | Phone Number |
|---|---|---|---|
| Jens Volkert | volkert | – | 888 |
| Siegfried Grabner | grabner | – | 884 |
| Dieter Kranzlmüller | dk | – | 885 |
| Alfred Spalt | spalt | – | 885 |
| Wolfgang Stürzlinger | wrzl | – | 884 |

### 4.5.1 nCUBE 2 Model 1

#### Footprint

| | |
|---|---|
| System | nCUBE 2 |
| Architecture | Distributed Memory MIMD |
| # of Processors | 64<br>4 IO Processors |
| Type of Processors | nCUBE Custom Chip |
| Memory Size | 4 MB per node |
| Disk Capacity | 4 GB |
| Operating System | Vertex, Vortex |
| Name | ncube00.edvz.uni-linz.ac.at |
| Internet Address | 140.78.3.11 |
| Location | UNI Linz |
| System Administrator | S. Grabner |
| Network Coordinator | S. Grabner |

### Availability

| Group | RISC | TU-Vienna | Vienna | GUP | Salzburg | World |
|---|---|---|---|---|---|---|
| | | | | good | | |

### Further Description

This machine is used for research and educational purpose.

### Footprint of the Host Computer

| | |
|---|---|
| System | SUN4/330 |
| Type of Processors | SPARC |
| Memory Size | 56 MB |
| Disk Capacity | 1.4 GB |
| Operating System | BSD 4.3 |
| Name | nCube.edvz.uni-linz.ac.at |
| Internet Address | 140.78.3.5 |
| Location | UNI Linz |
| System Administrator | S. Grabner |
| Network Coordinator | S. Grabner |

### Software

### NCC

- **Name:** ncc
- **Location:** /usr/ncube/current/bin
- **Type:** Crosscompiler for nCUBE 2 programs on SUN330
- **Characterization:** compiles C, FORTRAN, and Assembler code for the nCUBE on the SUN frontend. *ncc* is used like the normal *cc* compiler.
- **Functions:**
  - Invocation: `ncc`
  - Termination:
  - Help: `nman ncc`
  - Example: `ncc program.c`
  - Other important functions:
- **Installation:** include the path `/usr/ncube/current/bin`
- **Manuals:** nCUBE 2 Programmers Guide, Chapter 6

### XNC

- **Name:** xnc
- **Location:** /usr/ncube/current/bin/sun4
- **Type:** Loader for nCUBE 2 programs down to nCUBE nodes.
- **Characterization:**
- **Functions:**
  - Invocation: `xnc`
  - Termination: `nkill processID` – to terminate node programs
  - Help: `nman xnc`
  - Example: `xnc -d 4` *executable* – start a program on 16 nodes
  - Other important functions:
- **Installation:** include the path `/usr/ncube/current/bin/sun4` or `/usr/ncube/current/bin/‘arch‘`, respectively.
- **Manuals:** nCUBE 2 Programmers Reference (= nman page for *xnc*)

### NDB

- **Name:** ndb
- **Location:** /usr/ncube/current/bin/sun4
- **Type:** Symbolic source code and assmebly level debugger for nCUBE 2.
- **Characterization:**
- **Functions:**
  - Invocation: `ndb`
  - Termination: `quit`
  - Help: `nman ndb`
  - Example: `ndb -p processID` *executable* – if program is hung
  - Note: the source code must be compiled with ncc and the *-g* option.
- **Installation:** include the path `/usr/ncube/current/bin/sun4` or `/usr/ncube/current/bin/‘arch‘`, respectively.
- **Manuals:** nCUBE 2 Programmers Guide, Chapter 7

### XTOOL

- **Name:** xtool
- **Location:** /usr/ncube/current/bin/sun4
- **Type:** Execution Profiler for nCUBE 2 programs.
- **Characterization:** analyzes the execution profile of FORTRAN and C programs (graphically).
- **Functions:**
  - Invocation: `xtool` *programname*
  - Termination:
  - Help: `nman xtool or xtool "-?"`
  - Example:
  - Other important functions:
- **Installation:** include the path `/usr/ncube/current/bin/sun4` or `/usr/ncube/current/bin/‘arch‘`, respectively.
- **Manuals:** nCUBE 2 Programmers Reference, chapter 8

### CTOOL

- **Name:** ctool
- **Location:** /usr/ncube/current/bin/sun4
- **Type:** Communication Profiler for nCUBE 2 programs.
- **Characterization:** analyzes the communication profile of FORTRAN and C programs (graphically).
- **Functions:**
  - Invocation: `ctool` *options*
  - Termination:
  - Help: `nman ctool or ctool "-?"`
  - Example:
  - Other important functions:
- **Installation:** include the path `/usr/ncube/current/bin/sun4` or `/usr/ncube/current/bin/‘arch‘`, respectively.
- **Manuals:** nCUBE 2 Programmers Reference, chapter 9

## 4.6 Group Description of the Institute for Software Technology and Parallel Systems, University of Vienna (PPS)

### General Information about the Group

- **Group Name:** Research Group Prof. H.P. Zima
- **Group Leader:** Prof. H.P. Zima
- **Address:**

INSTITUT FÜR SOFTWARETECHNIK UND PARALLELE SYSTEME
UNIVERSITÄT WIEN
o.Univ.Prof.Dr. H. Zima

Brünner Str. 72
A-1210 Vienna, AUSTRIA

- **Phone Number:** ++43-1-392647/221 (Head)
- **Phone Number:** ++43-1-392647/222 (Secretary)
- **Phone Number:** ++43-1-40103/2790 (ACPC coordinator)
- **FAX Number:** ++43-1-392647/224
- **Internet domain name:** `par.univie.ac.at`
- **Bitnet domain name:** `awiuni11.bitnet`
- **Persons at Brünner Str. 72:**

| Name | Internet-account | Bitnet-account | Phone Number |
|---|---|---|---|
| Prof. Dr. H.P. Zima | zima | A4424DAJ | -221 |
| M. Morawitz (sec.) | sec | | -222 |
| E. Obermaier (sec.) | sec | | -222 |
| DI. S. Benkner | sigi | | -202 |
| Dr. Ing. R. Blasko | roman | | -226 |
| Dr. P. Brezany | brezany | | -227 |
| B. Chapman B.Sc.(Hons.) | barbara | | -205 |
| DI. T. Fahringer | tf | | -201 |
| DI. H. Moritsch | hm | | -202 |
| Dr. A. Schwald | schwald | | -201 |
| Dr. V. Sipkova | sipka | | -207 |
| E. Cikan (tech.) | cikan | | -206 |

- **Persons at Rathausstr. 19/II/3:**

| Name | Internet-account | Bitnet-account | Phone Number |
|---|---|---|---|
| DI. B. Knaus | bk | A4424DAI | ++43-1-40103-2790 |
| DI. P. Kutschera | peter | A4424DAK | ++43-1-40103-2789 |

### 4.6.1 Intel iPSC/860 Hypercube

#### Footprint

| | |
|---|---|
| System | Intel iPSC/860 |
| Architecture | MIMD Distributed Memory |
| # of Processors | 16 |
| Type of Processors | i860 |
| Memory Size | 8MB per processor |
| Disk Capacity | 1.3GB on CFS |
| Operating System | NX/2 |
| Name | cube |
| Internet Address | 131.130.70.34 |
| Location | Institut für Statistik und Informatik<br>Lehrkanzel Zima<br>Brünner Str. 72<br>1210 Wien |
| System Administrator | Barbara Chapman, barbara@par.univie.ac.at |
| Network Coordinator | |

#### Availability

| Group | RISC | TU-Vienna | Vienna | Linz | Salzburg | World |
|---|---|---|---|---|---|---|
| | y | y | y | y | y | n |

#### Further Description

#### Footprint of the Host Computer

| | |
|---|---|
| System | Frontend for Intel Hypercube |
| Type of Processors | Intel 80386 |
| Memory Size | |
| Disk Capacity | 380MB |
| Operating System | UNIX System 5, V 3.2 |
| Name | pixy |
| Internet Address | 131.130.70.33 |
| Location | Institut für Statistik und Informatik<br>Lehrkanzel Zima<br>Brünner Str. 72<br>1210 Wien |
| System Administrator | Barbara Chapman, barbara@par.univie.ac.at |
| Network Coordinator | |

**Software**

# PICL

- **Name:** Portable Instrumented Communication Library, Version 1.92
- **Location:** /newpicl (documentation in /picl-org)
- **Type:** Library package for portable parallel programming
- **Characterization:** PICL provides a library of routines which may be used instead of the native communication routines on any machine on which it is installed. Timestamped trace data is generated, automatically or as specified by the user (communication, idle times, events).
- **Software Description:** PICL provides portable syntax for parallel programs across several multiprocessor systems, including the Intel hypercubes and simulator, the Delta, the NCUBE and Cogent systems. It contains routines for communication primitives, such as *send* and *receive*, and related system calls, and widely-used higher routines such as global summing and broadcasts.

  It also has a built-in tracing facility, which may be controlled by the user. This enables the gathering of data on interprocessor communication, processor idle times, and user-defined events. A separate software package, ParaGraph, displays the tracing output graphically. This is currently installed on our local SUN network only. However, versions of ParaGraph are available for several types of workstation. Contact the system administrator for information on obtaining the source code.

  The most convenient way to use the PICL routines is to include the appropriate library at compile time. Separate libraries for the host and node code have been generated and are to be found in the directory /newpicl. Examples of usage of the routines are to be found in the User's Guide (see below).

Our installation comprises the library packages and the source code for the libraries in the subdirectories picl and port. Note that this version has been optimized for use on the iPSC/860. The original version of the PICL libraries is available in source form only in the directory /picl-org and its subdirectories.

- **Installation:**

  The appropriate libraries should be linked in when the host and node programs are compiled. A description of how to do this is to be found in README-USAGE.

- **Manuals:** There are two manuals. The user's guide is located in userguide.tex (and .sty). The C reference manual is provided in cref.tex. Both are located in the directory /picl-org. These document the original version of PICL. The modifications are described in the two upgrade files in /newpicl. Note that there is currently no Fortran reference manual.

### 4.6.2 Transputersystem

#### Footprint

| | |
|---|---|
| System | Impuls 2400 |
| Architecture | Transputersystem |
| # of Processors | 16 |
| Type of Processors | T800 |
| Memory Size | 1 MByte |
| Disk Capacity | 0 |
| Operating System | iserver |
| Name | — |
| Internet Address | — |
| Location | Rathausstr. 19/2/3<br>A-1010 Wien |
| System Administrator | Peter Kutschera (peter@merlin.par.univie.ac.at) |
| Network Coordinator | |

#### Availability

| Group | RISC | TU-Vienna | PPS | Vienna | Linz | Salzburg | World |
|---|---|---|---|---|---|---|---|
| | y | y | y | y | y | y | y |

To get an account ask peter@merlin.par.univie.ac.at .

#### Further Description

#### Footprint of the Host Computer

| | |
|---|---|
| System | HP9000/425s |
| Type of Processor | 68040 |
| Memory Size | 16 MB |
| Disk Capacity | 1.2 GB |
| Operating System | Domain/OS 10.3.5 |
| Name | merlin.par.univie.ac.at |
| Internet Address | 131.130.34.65 |
| Location | Rathausstr. 19/2/3<br>A-1010 Wien |
| System Administrator | Peter Kutschera (peter@merlin.par.univie.ac.at) |
| Network Coordinator | |

The system is also connected to an PC (no access from the network).

**Software**

**C–Compiler**

- **Name:** INMOS C compiler Version 2.01.03 (built at 17:51:52 Oct 16 1990)
- **Location:** icc and tools are in **/usr/local/bin**
- **Type:** Compiler, linker,...
- **Characterization:** Takes C-sources with library-calls for communication and a description of the configuration of the transputer–system and creates a bootable file. This file can be executed on the transputers.
- **Functions:** NO automatic parallelization or communication
  - Invocation: `icc`
  - Termination: `none`
  - Help: `manuals`
  - Example: `see /usr/local/ictools/examples`
  - Other important functions:
- **Installation:** All used environment-vars are automatically set to defaults.
- **Manuals:** ANSI C Toolset User Manual, INMOS Limited
  ANSI C Toolset Language Reference, INMOS Limited

### Occam2–Compiler

- **Name:** occam 2 compiler Version 2.01.01 (21:15:57 Feb 27 1991) (Transputer)
- **Location:** oc and tools are in **/usr/local/bin**
- **Type:** Compiler, linker,...
- **Functions:**
  - Invocation: `oc`
  - Termination: `none`
  - Help: `Manuals`
  - Example: `see /usr/local/d7205/examples`
  - Other important functions:
- **Installation:** All used environment-vars are automatically set to defaults.
- **Manuals:** OCCAM User Manual Part 1: User Guide and Tools, INMOS Limited
  OCCAM User Manual Part 2: Occam Libraries, INMOS Limited

### config

- **Name:** config, Version 0.2
- **Location:** **/usr/local/bin**
- **Type:** Utilitiy
- **Characterization:** Tool to set or determine the configuration of the links between the transputers.
- **Functions:**
  - Invocation: `config options`
  - Help: `config`
  - Example:

```
config -net 4 4           # set link-switches
config -show              # check configuration
config -cr_cfs > test.cfs # create Hardware-Part of .cfs-file
```

- **Installation:** none
- **Manuals:** none

### 4.6.3 MasPar

#### Footprint

| | |
|---|---|
| System | MasPar MP-1101A |
| Architecture | Massively parallel SIMD |
| # of Processors | Data Parallel Unit (DPU)<br>with 1024 Processor Elements (PE) |
| Type of Processors | PE: full-custom processor chip (1.8 MIPS) |
| Memory Size | DPU: 16 KBytes in each PE, 1.128 MBytes in ACU |
| Disk Capacity | — |
| Operating System | — |
| Name | — |
| Internet Address | — |
| Location | Institut fuer Statistik und Informatik |
| | Lehrkanzel Zima<br>Bruenner Str. 72, 1210 Wien |
| System Administrator | Viera Sipkova sipka@par.univie.ac.at |
| Network Coordinator | |

#### Availability

| Group | RISC | TU-Vienna | Vienna | Linz | Salzburg | World |
|---|---|---|---|---|---|---|
| | | | | | | |

#### Further Description

MasPar MP-1 is a massively parallel, data-parallel, fine-grained processing system. It consists of 3 Subsystems: Front-End (FE), Data Parallel Unit (DPU), and I/O Subsystem. The FE is a DEC 5000 workstation which runs UNIX operating system with graphics and network services, including the software environment in which all the MasPar tools and utilities are executed. The DPU performs all parallel processing. It is composed of the Array Control Unit (ACU), Processor Element Array (PE Array) and Communication Mechanisms. The ACU is a register- based (14 MIPS) processor with its own registers, data memory (128 KBytes), and instruction memory (1 MByte - 4 GBytes). The PE array consits of 1K PEs that are arranged in a 2-dimnesional matrix (32x32). Each PE contains the arithmetic- logical unit (1.8 MIPS processor), registers, and 16 KBytes of RAM. Communications are implemented via VME Bus, ACU-PE Bus, and X-Net and Global Router.

Peak performance of MasPar MP-1: Each PE operates at 1.6 MIPS (32-bit integer add) and 80Kflops (average of 32-bit floating point multiply and add). MP-1 systems currently support PE arrays with 1,024 to 16,384 processor elements. Peak performance ranges from 1,600 to 26,000 integer MIPS (32-bit add), from 82 to 1,300 single precision Mflops (average of add and multiply) and from 36 to 580 double precission (64-bit) Mflops.

**Footprint of the Host Computer**

| | |
|---|---|
| System | DEC Station 5000 |
| Type of Processors | |
| Memory Size | 24 MB |
| Disk Capacity | 665 MByte hard disk<br>optical compact disc drive (635 MByte) |
| Operating System | ULTRIX |
| Name | maspar |
| Internet Address | 131.130.70.36 |
| Location | Institut fuer Statistik und Informatik |
| | Lehrkanzel Zima<br>Bruenner Str. 72, 1210 Wien |
| System Administrator | Viera Sipkova sipka@par.univie.ac.at |
| Network Coordinator | |

**Software**

**ULTRIX and Misc. Tools**

- **Name:** ULTRIX Worksystem Software, Version 4.0 (ULTRIX Operating System; DEC-Windows System);
  MasPar System Software Version 2.1, Version 2.2
- **Location:**
- **Type:** Programming languages: MPL (MasPar Application Language), MPF (MasPar Fortran);
  Compilers: mpl, mpfortran, mpl_cc, mpl_fort;
  Programming environment MPPE.
- **Characterization:** The mpl language is based on K&R C and is used for programming the DPU. The MPF language is based on FORTRAN 77 with extensions from DEC-s VAX Fortran and the new Fortran 90 array statements. The MPPE is a set of tools for running, debugging and analysing of parallel programs.
- **Functions:** Compilers and the MPPE are used in a way which is standard for UNIX operating system and DEC-windows.
  - Invocation: `in standard way for UNIX and DEC-windows`
  - Termination: `in standard way for UNIX and DEC-windows`
  - Help: `by online manuals`

  - Example: `mpfortran -o exam exam.f`
  - Other important functions:

- **Installation:** setenv $MP\_PATH/usr/maspar$;
  $setpath$MP_PATH/bin

- **Manuals:** MP-1 System Hardware manuals Version 2.1,
  MasPar Overview and MPPE manuals Version 2.1, Version 2.2 ,
  MasPar MPL Manuals Version 2.1, Version 2.2,
  MasPar Fortran Programming Manuals Version 2.1,
  ULTRIX Manuals Version 4.0

  Location: Institut fuer Statistik und Informatik, Bruenner Str. 72, 1210 Wien

## 4.7 Group Description of RIST++ Salzburg

### General Information about the Group

- **Group Name:** Research Institute for Software Technology + PLUS (Paris Lodron University Salzburg) (RIST++ -Salzburg).
- **Group Leader:** Prof. Dr. Peter Zinterhof.
- **Address:**

  RIST++
  Research Institute for Software Technology
  Paris Lodron University Salzburg
  Hellbrunnerstr. 34
  A-5020 Salzburg, Austria, Europe

  RIST++
  Research Institute for Software Technology
  Paris Lodron University Salzburg
  Jakob Haringerstr. 5
  A-5020 Salzburg, Austria, Europe

- **Phone Number:** 0662-8044 (secretary: 0662-8044-6715).
- **FAX Number:** 0662-8044-5010
- **Persons at Hellbrunnerstr. 34:**
  **Domain Name:** Internet: `EDVZ.SBG.AC.AT`

| Name | Internet-account | Bitnet-account | Phone Number |
|---|---|---|---|
| Prof. Dr. P. Zinterhof | ZINTERHOF | | 6714 |
| Christine Huber (sec.) | HUBERC | | 6715 |
| Mag. Erika Hausenblas | HAUSENBLAS | | 5308 |
| Mag. Andrea Lauß | LAUSS | | 5323 |
| Wolfgang Schmid | SCHMID | | 5308 |
| Stefan Wegenkittl | | | 5308 |
| Peter Zinterhof jun. | ZINTER2 | | 6727 |

- **Persons at Jakob Haringerstr. 5:**
  **Domain Name:** Internet: `kirk.coma.sbg.ac.at`

| Name | Internet-account | Bitnet-account | Phone Number |
|---|---|---|---|
| Ing. Ernst Forsthofer | forsthof | | 6770 |
| Dr. Norbert Ulamec | ulamec | | 6770 |
| Mag. Markus Salchegger | salchegg | | 6771 |
| Peter Schmitzberger | schmitz | | 6771 |

### 4.7.1 SUN 3/60

**Footprint**

| | |
|---|---|
| System | SUN 3/60 |
| Architecture | Single Processor |
| # of Processors | 1 |
| Type of Processors | 68020 |
| Memory Size | 12 MB |
| Disk Capacity | 1 GB |
| Operating System | SUN OS 4.1 |
| Name | kirk.coma.sbg.ac.at |
| Internet Address | 141.201.6.2 |
| Location | RIST++, |
| | Uni Salzburg |
| System Administrator | E. Forsthofer |
| Network Coordinator | E. Forsthofer |

**Availability**

| Group | RISC | TU-Vienna | Vienna | Linz | Salzburg | World |
|---|---|---|---|---|---|---|
| | y | y | y | y | y | y |

**Software**

**STRAND$^{\underline{88}}$**

- **Name:** STRAND$^{\underline{88}}$ Buckingham Release
- **Location:** /usr/local/strand
- **Type:** Parallel Declarative Programming Language
- **Characterization:** STRAND$^{\underline{88}}$ is an implementation of the Strand language, which includes an environment for developing and running Strand programs. The Strand language is desinged to utilize the facilities of concurrent machines and parallel architectures. It supports concurrent processing, stream communication between processes, Data-flow, modular development, declarative reading, object oriented programming and interfaces to other languages and data types.
- **Functions:**
  - Invocation: `st`
  - Termination: `<CTRL>C`
  - Help: `st -h`

- Example:
- Other important functions:

- **Installation:** `source /usr/local/strand/strandrc`
- **Manuals:** STRAND88 User Manual,
  Book: Strand New Concepts in Parallel Programming

### 4.7.2 SPARC 2

#### Footprint

| | |
|---|---|
| System | Sparc 2 |
| Architecture | Single Processor |
| # of Processors | 1 |
| Type of Processors | SPARC |
| Memory Size | 32 MB |
| Disk Capacity | 420 MB |
| Operating System | SUN OS 4.1.1 |
| Name | spock.coma.sbg.ac.at |
| Internet Address | 141.201.6.3 |
| Location | RIST++, |
| | Uni Salzburg |
| System Administrator | E. Forsthofer |
| Network Coordinator | E. Forsthofer |

#### Availability

| Group | RISC | TU-Vienna | Vienna | Linz | Salzburg | World |
|---|---|---|---|---|---|---|
| | y | y | y | y | y | y |

#### Software

#### STRAND88

### 4.7.3 SPARC ELC

#### Footprint

| | |
|---|---|
| System | Sparc ELC |
| Architecture | Single Processor |
| # of Processors | 1 |
| Type of Processors | SPARC |
| Memory Size | 8 MB |
| Disk Capacity | 0 MB |
| Operating System | SUN OS 4.1.1 |
| Name | scotty.coma.sbg.ac.at |
| Internet Address | 141.201.6.4 |
| Location | RIST++, |
| | Uni Salzburg |
| System Administrator | E. Forsthofer |
| Network Coordinator | E. Forsthofer |

#### Availability

| Group | RISC | TU-Vienna | Vienna | Linz | Salzburg | World |
|---|---|---|---|---|---|---|
| | y | y | y | y | y | y |

#### Software

#### STRAND88

### 4.7.4 CONVEX C220

#### Footprint

| | |
|---|---|
| System | CONVEX C220 |
| Architecture | Parallel Vector |
| # of Processors | 2 |
| Type of Processors | Vector-processor |
| Memory Size | 128 MB |
| Disk Capacity | 8 GB |
| Operating System | CONVEX OS 9.1 |
| Name | CONVEX |
| Internet Address | 141.201.10.2 |
| Location | TECH Consult |
| | Salzburg |
| System Administrator | Kurt Krenn |
| Network Coordinator | Kurt Krenn |

#### Availability

| Group | RISC | TU-Vienna | Vienna | Linz | Salzburg | World |
|---|---|---|---|---|---|---|
| | | | | | good | |

#### Further Description

#### Software

#### Vectorizing C-Compiler

- **Name:** CONVEX Vectorizing C-CompilerV 2.0
- **Location:**
- **Type:** Parallelizing and Vectorizing Compiler
- **Characterization:**
- **Hints:** To see, how to use the HIGH PERFORMANCE MATHEMATICAL VECTOR LIBRARY (VECLIB) type: man veclib. There are many vectorized functions (subroutines) you can use in your application-programs.
- **Functions:**

- Invocation: `cc`
- Termination: `<CTRL>C`
- Help: `man cc`
- Example:
- Other important functions:

- **Installation:**
- **Manuals:** Vectorizing C-Compiler User Manual,

### 4.7.5 ALPHA Cluster (6+1 DEC 3000 Alpha AXP 400)

**Footprint**

| | |
|---|---|
| System | 1$\times$ DEC 3000 Alpha AXP 400 |
| Architecture | Single Processor |
| # of Processors | 1 |
| Type of Processors | DECchip 21064/133 MHz |
| Memory Size | 64 MB |
| Disk Capacity | 1 GB |
| Operating System | DEC OSF/1 T1.2-2 |
| Name | picard.coma.sbg.ac.at |
| Internet Address | 141.201.65.2 |
| Location | RIST++, |
| | Uni Salzburg |
| System Administrator | E. Forsthofer |
| Network Coordinator | E. Forsthofer |

| | |
|---|---|
| System | 6× DEC 3000 Alpha AXP 400 |
| Architecture | Single Processor |
| # of Processors | 1 |
| Type of Processors | DECchip 21064/133 MHz |
| Memory Size | 32 MB |
| Disk Capacity | 1 GB |
| Operating System | DEC OSF/1 T1.2-2 |
| **Names and addresses** | |
| Name | riker.coma.sbg.ac.at |
| Internet Address | 141.201.65.6 |
| Name | laforge.coma.sbg.ac.at |
| Internet Address | 141.201.65.7 |
| Name | worf.coma.sbg.ac.at |
| Internet Address | 141.201.65.8 |
| Name | crusher.coma.sbg.ac.at |
| Internet Address | 141.201.65.9 |
| Name | data.coma.sbg.ac.at |
| Internet Address | 141.201.65.10 |
| Name | troi.coma.sbg.ac.at |
| Internet Address | 141.201.65.11 |
| Location | RIST++, |
| | Uni Salzburg |
| System Administrator | E. Forsthofer |
| Network Coordinator | E. Forsthofer |

**Availability**

| Group | RISC | TU-Vienna | Vienna | Linz | Salzburg | World |
|---|---|---|---|---|---|---|
| | y | y | y | y | y | y |

**Software**

**DEC C++ for DEC OSF/1 AXP**

**DEC FUSE for DEC OSF/1 AXP**

**DEC Fortran for DEC OSF/1 AXP**

# Appendix A

# Table of All Available Parallel Machines in the ACPC Network

This appendix can be seen as a summary of Chapter 4 as it shows all parallel machines in the ACPC Network in just one table. The information given in this table is sufficient to set up a connection with any machine in the ACPC Network (even if there is no name service available). However, if one wants to know details of the machines mentioned in Table A.1 one has to look for the corresponding section in Chapter 4.

As several machines like the Transputers and the nCube can only be used from a host machine, for these machines this host is put into Table A.1.

| Group | Type of Machine | Symbolic Address | Numeric Address |
|---|---|---|---|
| RISC-Linz | Sequent Symmetry | melmac.risc.uni-linz.ac.at | 193.170.36.100 |
| | Transputer | venus.risc.uni-linz.ac.at[1] | 193.170.36.70 |
| Uni-Vienna | Transputer | epikur.ani.univie.ac.at[1] | 131.130.32.113 |
| TU-Vienna | Sequent Balance | eacpc1.tuwien.ac.at | 128.130.36.3 |
| | Intel Hypercube | eacpc2.tuwien.ac.at[1] | 128.130.36.4 |
| GUP-Linz | nCube | nCube.edvz.uni-linz.ac.at[1] | 140.78.3.5 |
| Uni-Vienna | Intel Hypercube | pixy.par.univie.ac.at[1] | 131.130.70.33 |
| (PPS) | MasPar | maspar.par.univie.ac.at[1] | 131.130.70.36 |
| | Transputer | merlin.par.univie.ac.at[1] | 131.130.34.65 |
| RIST++ | Alpha Cluster | picard.coma.sbg.ac.at | 141.201.65.2 |
| (Salzburg) | Convex | convex.edvz.sbg.ac.at | 141.201.10.2 |

Table A.1: List of All Parallel Machines in the ACPC Network

[1] This is the address of the host computer that has to be used to get access to the parallel machine.

# Appendix B

# Alphabetical List of Persons in the ACPC

This appendix summarizes the most important information about persons that are working in the ACPC by associating with each person her/his email address and telephone number. More information for each person can be found in the appropriate group description of Chapter 4.

# *

Index