Par.C
SYSTEM

Easy access to 100% of the transputer's power

Dynamic vs. static parallelism

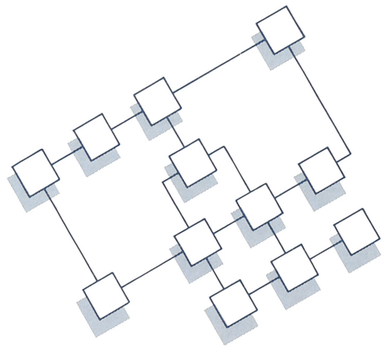
Parsec

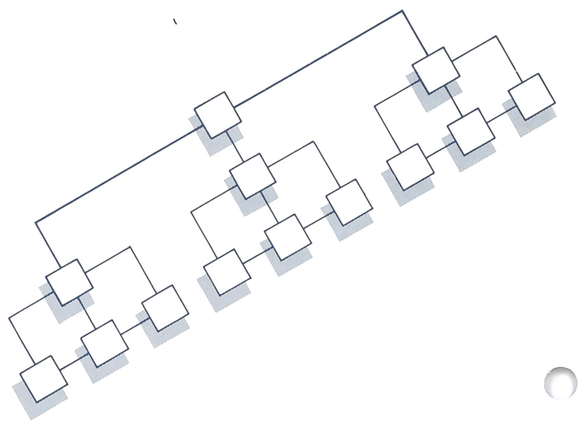
Dynamic parallelism

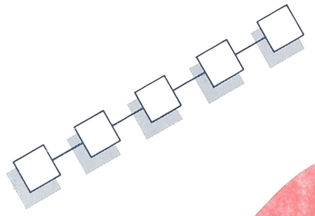
Static parallel systems require the number of processes and their interconnections to be fixed and known at compiletime. A dynamic parallel system postpones the decisions on the parallel control flow of the program until it is actually run. Only such a system allows the development of programs which are completely independent of the transputer systems on which they will run.

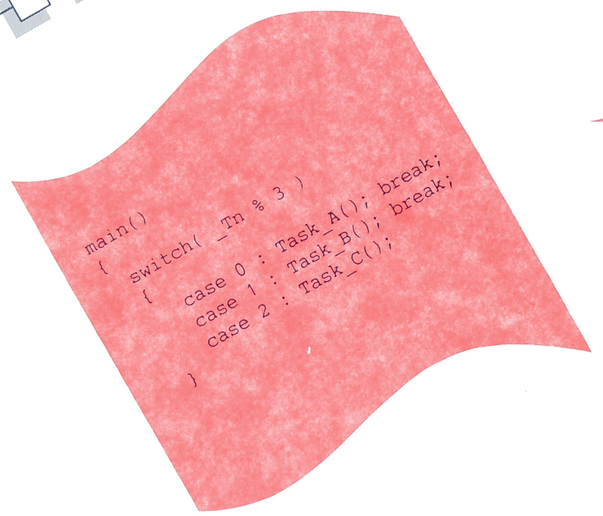
Imagine a program which runs on any cluster of transputers, regardless of the network topology or size, the transputer types and the available memory on each node. The ideal of ''plug in an extra board and your program runs faster'' can now actually be realised.

The Par.C System offers the software that is needed to make full use of the flexibility of the transputer.









Example

In this oversimplified example program, each node's identity, available in the `_Tn` variable, is used to determine which of the three tasks is executed on that node. No matter what the size of the transputer network during each run, the three tasks will be more or less equally divided over the same amount of worker transputers.

More than just a compiler

The Par.C System is more than yet another C compiler for transputers. A compiler merely generates code. The Par.C System offers much more:

– Complete transparency and consistency without conflicts regarding parallel processes, recursive functions, event, signal, exit and abort handling, file I/O, mixed priority levels, channel communication, semaphores, on-chip memory usage.

– All the support to produce flexible parallel programs in an easy way. The runtime system includes memory management, multi-process file I/O and all ANSI C functions in re-entrant versions plus a number of special transputer functions.

Cost-effective

The Par.C System has been designed to decrease the cost of software development for transputer systems. Moreover, maintenance of this software will also be less expensive. Programs do not have to be changed when changing the network topology or size: they can simply be loaded again after the hardware has been changed.

Simple concepts yielding complex possibilities

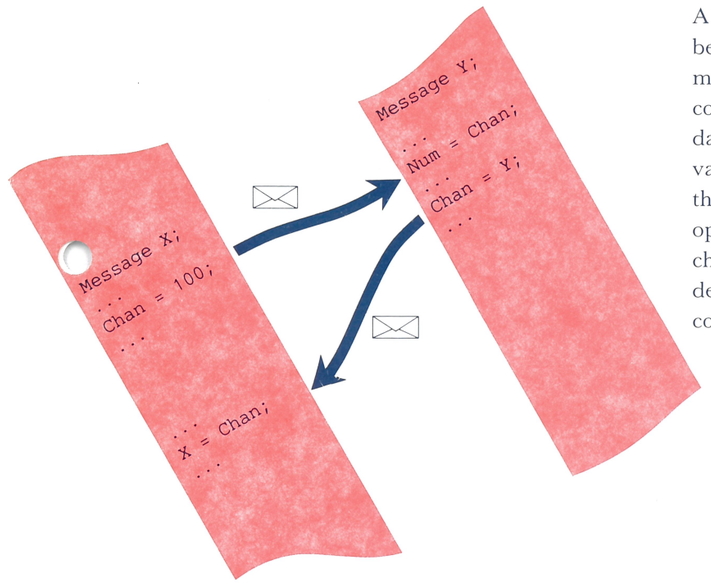
Parsec

Language extensions

The Par.C System is based on a superset of the standard C programming language. The language extensions are few and very powerful. They reflect the elementary functions needed in parallel programming:

- starting concurrent processes
- inter-process communication
- awaiting the first of a number of possible events

By introducing these basic concepts and combining them with the flexibility which is common to standard C, a wide range of possibilities becomes available: from programs with static structures to software with a structure which adapts itself to real-time demands and the resources available at runtime.

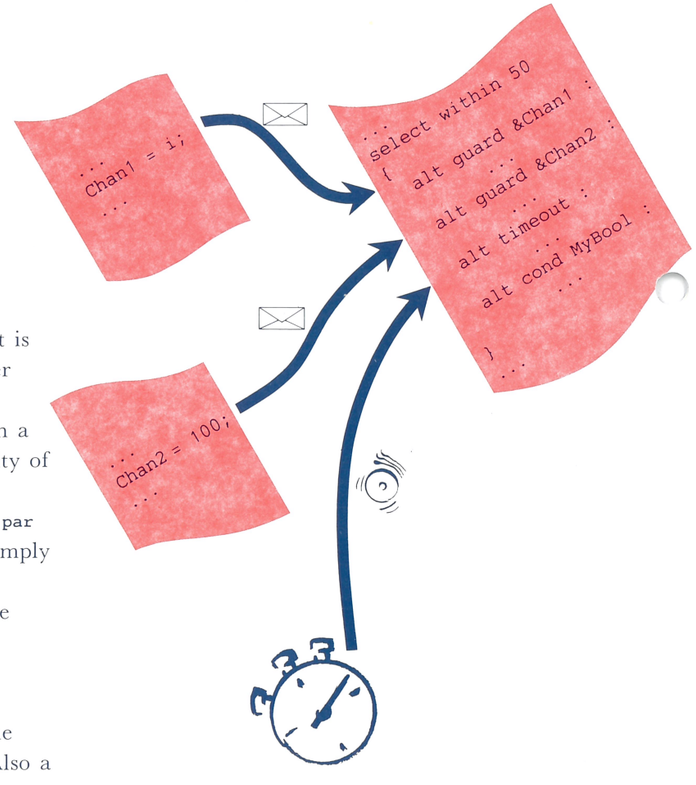


Channel

A `channel` is a point-to-point connection between two processes. Both processes must have arrived at the corresponding communication statements before the datatransfer takes place. Any type of variable or structure can be transmitted through a `channel` using the assignment operator. The conventional strong type checking of C will enhance correct definition and implementation of communication protocols.

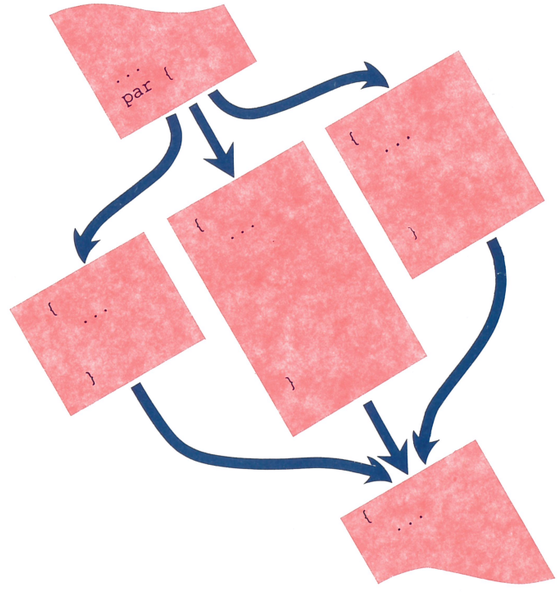
Select

The `select` construct resembles the conventional C switch construct, but is far more powerful: conditions, rather than constants are evaluated. Each alternative (cf. case) can also contain a `guard`, which monitors the availability of input on a channel. Conditions and guards can be replicated like in the `par` statement. Priorities are indicated simply by the evaluation sequence of guards/conditions. When none of the conditions evaluates to `TRUE`, the process will wait until at least one guarded channel will have input available, and then proceeds with the code specified for that alternative. Also a `timeout` value can be specified.



Par

Each (compound) statement inside the `par` body is executed in parallel with the others, and can contain any code. The process executing the `par` awaits termination of the concurrent processes. The header of the `par` may contain any replicator of the form used in a for statement. The number of replications of the statements within the `par` body can therefore be dynamically decided.

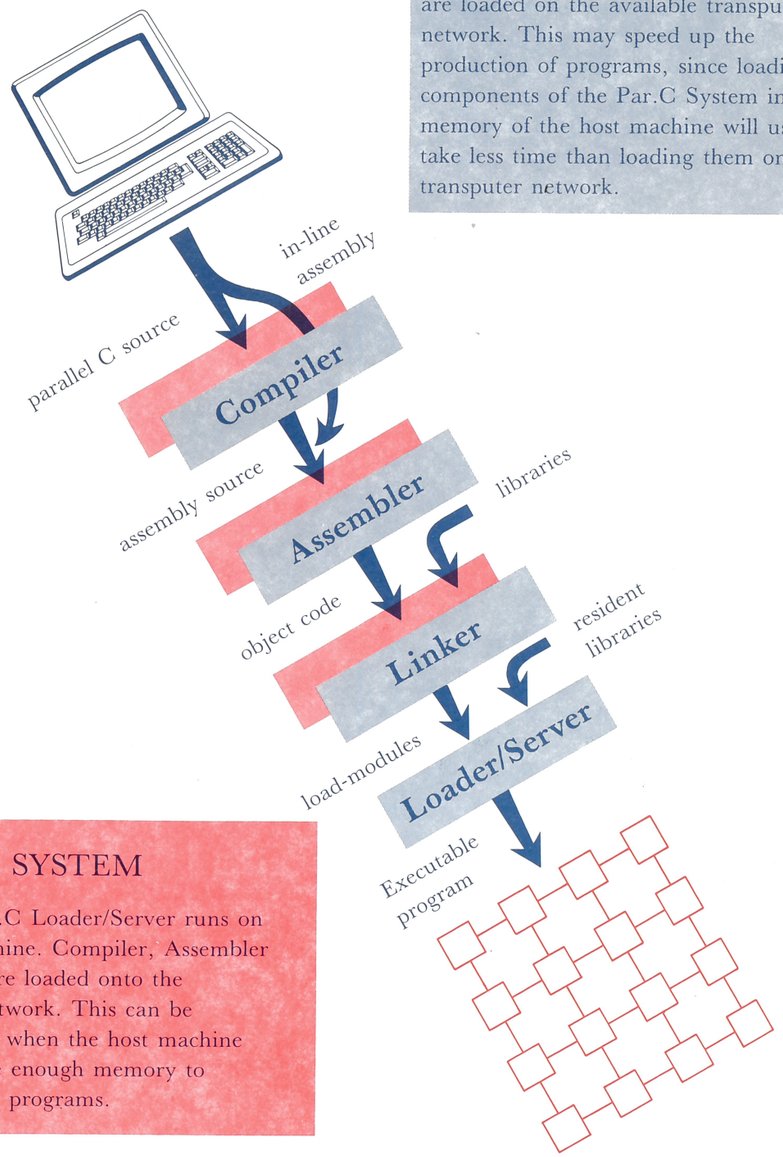


Your best solution: the Par.C System

Parsec

Cross system

The Compiler, Assembler, Linker and Loader/Server all run on the host machine. Only Par.C-produced programs are loaded on the available transputer network. This may speed up the production of programs, since loading the components of the Par.C System into the memory of the host machine will usually take less time than loading them onto the transputer network.



Native system

Only the Par.C Loader/Server runs on the host machine. Compiler, Assembler and Linker are loaded onto the transputer network. This can be advantageous when the host machine does not have enough memory to produce large programs.

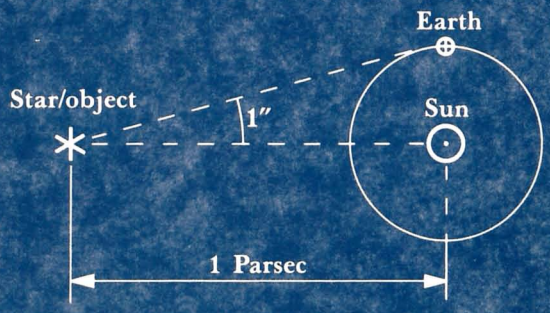
Features of the Par.C System

- ***Easy to use:***
  - Few and powerful language extensions, implemented in accordance with standard C.
  - No separate configuration file or language needed.
  - Fully re-entrant file I/O system.
- ***Complete:***
  - Full ANSI-C runtime libraries.
  - Access to all features of the transputer's hardware.
- ***Flexible:***
  - Programs can be loaded into any transputer network.
  - Language extensions allow any kind of dynamic parallel structure.
- ***Cost-effective:***
  - Low software development costs through ease-of-use and flexibility.
  - Low software maintenance costs through adaptability of the software to changes in the hardware.

Availability

The Par.C System is available in two different versions, each of which can be delivered for a range of host machines. Native versions are avaialable for IBM-PC, SUN workstations, Harris HCX mainframes, and transputers running MultiTool or Helios. Cross system versions are available for IBM-PC, SUN workstations and Harris HCX mainframes. Other host machines will be supported in the near future.

Par.C SYSTEM



Parsec, the unit in which distances of stars are measured. The distance of an object to the sun corresponds to one parsec, if the distance between the sun and the earth, as viewed from this object, has an angular size of one second of arc. One parsec is approximately 19 million million ($1.9 * 10^{13}$) miles, or 3.26 light years.

Parsec Developments P.O. Box 782 2300 AT Leiden The Netherlands Tel. + 31 71 142142 Fax. + 31 71 134449