

USING DS 3000/1000 WITH HP 1000 MASTER PROGRAMS

JOERG MUELLER

J. MUELLER
SWS SOFTWARE SYSTEMS
BERN, SWITZERLAND

by Joerg Mueller

SWS SoftWare Systems

Bern, Switzerland

Introduction

HP/3000 is a very efficient general purpose computer, which gives users and programers a lot of flexibility on the software side. Now that newer HP/3000 models support HP-IB devices flexibility has been extended to the hardware. Nevertheless users would be ill advised to use HP/3000 as a measurement data logger or to do any other form of fast data acquisition. Interrupts at the rate of thousands or even only hundreds per second should not be sent to a general purpose computer.

Of course Hewlett Packard offers a solution for fast data acquisition and processing with the HP/1000 computers. An important additional advantage of the HP/1000 systems is a price tag which favourably compares to the /3000-line. Users already owning an HP/3000 see the disadvantage though of a largely incompatible, different 16 bit computer system. Architecture, operating system, language syntax and file structure are all different on the two system types. Except for trivial cases it is not even possible to define a common Fortran subset, so that the same source code could be used on both computers. The choice of having both types of systems within one company may nevertheless pay off, because a good price to performance ratio can be achieved. Once the decision has been made to purchase both HP/3000-

and HP/1000-computers it is obviously attractive to link the systems with a DS3000-1000 communication line. In fact in certain cases it may even be possible to have a memory resident HP/1000 operating system without any mass storage of its own thanks to the DS3000-1000 downloading facility.

The HP/1000 systems have evolved and this allows to use them also as general purpose computers. Nevertheless it is advisable to restrict the use of the /1000 and to do most of the programming on the HP/3000, if the resources such as the present workload and the communication line allow it, because software maintenance on the HP/3000 often is easier and has to be done for other applications anyway, so that the programmer staff usually will be more familiar with the /3000. However the environment may impose a more important HP/1000 configuration. Some problems with DS3000-1000 which may occur under such circumstances will here be discussed. Program to program communication (PTOP) will be analysed primarily in this paper, but the other forms of communication also will be discussed briefly so as to see their respective merits.

1. Respective merits of the DS3000-1000 features

1.1 Remote operator commands

Most of the RTE-IVB operator commands can be issued from a HP/3000 remote session. It is possible to start and analyse the state of the RTE-system. In fact a RTE-IVE memory based system may entirely be controlled by a HP/3000 session. The HP/1000 is working as a front end computer in such a case, a configuration, which may be very useful.

It is interesting to note that on the HP/3000 a Cross-Assembler of the HP/2100 is available (which in effect allows a subset of the HP/1000 instructions) and a Cross Loader also exists. A Fortran crosscompiler exists in the contributed library now. It therefor now should be possible to develop a complete HP/1000 application on the /3000 and simply download by DS3000-1000. The HP/1000 does not need to have any storage device in such a case.

1.2. Remote EXEC calls

An EXEC call is familiar to all HP/1000 programmers. The HP/3000 equivalent are the system intrinsic calls. The system resources may programatically be used by such calls. Input/Output, program scheduling, segment loads and time requests are possible. It is therefore feasible for a HP/3000 process to use all HP/1000 peripherals. Only the actual driver of the device is residing on the HP/1000 in such a case.

The opposite type of calls, i.e. the use of HP/3000 system intrinsics by a HP/1000 program, is not possible however.

1.3. Remote file access

Although HP/1000 files are not easily accessible by HP/3000 session commands, it is however fully possible to use HP/1000 files by a HP/3000 program. Such files cannot only be read and written to, but they can be created, opened and closed, renamed or also purged. The structure of the files remain of the HP/1000 type of course.

Besides different file structures the two computer systems also have different real number representation. This must be remembered for binary files which are commonly used by HP/1000 and HP/3000 programs. Except for a little snag in case of a zero the SPL-routines CR1R3 and CR3R1 contained in the contributed library allow correct conversion between the two types of real variables.

Again it is not possible for a HP/1000 program to access HP/3000 files. One obvious problem in this context is the maximum length of a name, which may only have up to six characters for a file on the HP/1000 instead of 8 letters as on the /3000.

1.4 Program to Program Communication (PTOP)

The most powerful communication tool available within DS3000-1000 is the PTOP-package. It consists of a set of nearly identical procedures available on both computers.

For a PTOP user application to work a pair of programs must exist. The program initiating the communication is the so called master program. The other program is called a slave program and is automatically scheduled by the respective operating system, when the master program initiates the communication. The master program may be on the HP/3000 or on the HP/1000. Since a HP/3000 program can practically control all resources of the HP/1000, it will not often be necessary to have HP/3000 master programs. Usually the other forms of remote access will be sufficient. Whenever a HP/1000 program wants to access the /3000 there is no other choice however than to use PTOP communication.

The master program initiates communication by the POPEN procedure. The slave receives all information by a GET call and returns its information by ACCEPT or REJECT. On the master side the following other procedures are available:

PREAD	to receive a buffer of information,
PWRIT(E)	to send a buffer of information,
PCONT(ROL)	to exchange a "tag"-field i.e. a buffer of 20 words,
PCLOS(E)	to abort the slave program,
PCHECK (/3000)	to analyse the last PTOP transaction.

Although the procedure calls are very simple, PTOP communication can present some difficulties to a beginner. Whenever multiple master programs initiate the same slave or automatic scheduling is wanted, problems may increase even more. But on the other hand applications can be tackled, which otherwise could not be realised at all. In fact once DS3000 is installed, PTOP can prove to be a very useful tool for concurrently running programs within the same computer system.

2. Situation of CIBA-GEIGY Photochemie (Fribourg, Switzerland)

The computer configurations to be discussed here as a practical case have the following simplified outline (Fig.1).

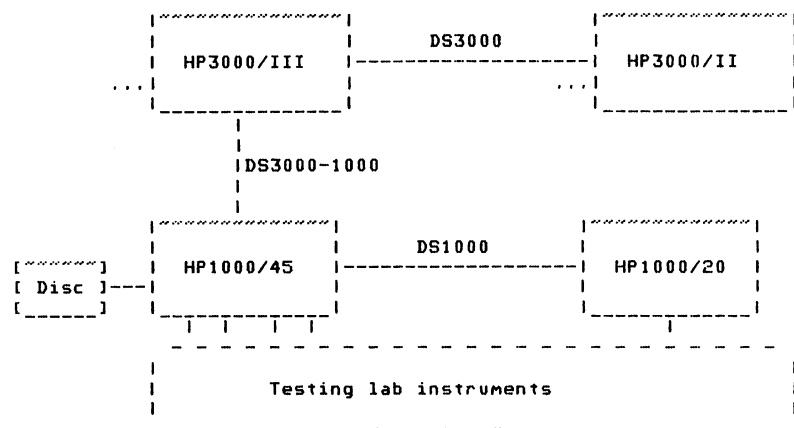


Fig. 1
Configuration with hardwired
connections.

The HP3000/II system serves primarily as development system, the HP3000/III supports materials control, production management, quality control and distribution and is thereby fully loaded. The HP/1000 systems perform measurements, control the instruments and process the data for Quality Control. The results are intermediately stored on the HP/1000 disc and usually daily transmitted to a HP/3000 data base.

Had the HP/3000 systems not been so heavily loaded, a configuration without disc-system could have been envisaged. Since this solution was out of the question, the benefits of largely independent HP/1000 systems were fully exploited. System operation still had to be simple though since no actual computer operator could be afforded for the /1000-systems.

It is here that local PTOP processing on the HP/1000 systems proved to be very useful. Concurrently running programs could thereby easily be synchronised. The PTOP programs running with the HP/3000 are not of such a nature as shall be shown later.

The usual contact between the HP1000/45 and the HP3000/III during the normal working hours restricts itself to a few occasional /3000-sessions initiated on the /1000 or the fetching of a source file as will be seen later. The daily evening transmission sends all relevant measurement results for complete evaluation and storage to the /3000. Since the /3000 systems are continuously operator controlled this is also the place all plotting is done through batch jobs streamed by the transmission program.

The transmission programs contain automatic maintenance functions so as to eliminate measurement results older than two days on the

/1000-disc, if transmission and storage on the /3000 were faultless.

The remaining important function of the DS 3000-1000 line enables to set up the HP/1000 master data files from scratch. This is done after every disc copy (usually once a month), whenever a new software revision is installed and when the master sets are updated on the /3000 side. The only updating mechanism for those /3000 master sets allowed to the user go by way of the /1000. Thereby data integrity usually is conserved eventhough the information is stored on both systems. Of course there always will be a hurried smart user ruining the concept.

Except for the occasional problems with ignorant new users the system has been successfully operational in automatic mode for several months now.

3. DS3000-1000 PTOP programs

For the three types of transmission operation mentioned three pairs of PTOP programs had to be written:

- 1) PUGE: to put or get source files
- 2) PUTSE: to transmit and maintain
the measurement result files
- 3) FMDEN: to maintain the master data sets

All three applications obviously aim to do programatic remote file access from the /1000 to the /3000. As it was shown earlier this function only can be achieved by PTOP communication. Had the amount of initial problems been foreseen it probably would have been more economical to sacrifice the requirement of at least semi-automatic operation and have the user manually start a /3000 session and a /3000 program, which then could directly do remote file access without PTOP communication. Now the PTOP pairs are operational they offer easier use though.

3.1. PUGE (PUT and GET)

Source file maintenance, i.e. creating back ups and only keeping source files, which are being worked on directly on disc, is a housekeeping problem common to all computer systems. Having DS3000-1000 this problem could elegantly be solved with some

additional benefits.

The programmers disc cartridges containing his source files are declared a scratch area and are usually entirely erased every week or two. Whereas such a way of operating a HP/3000 probably would create a programmers revolution, the HP/1000 with a DS-link can be maintained in such a manner, because

```
:RU,PUGE
+GET source
```

will rapidly give the file back (it takes 8 seconds to get the + prompt and e.g. 25 seconds for a source file of 300 lines to be available on the /1000). Of course eventually the source files are not available on line on the /3000 either anymore. In that case the programmer has to have the source reloaded from mag tape on to the /3000 and can then fetch it again, obviously a more tedious operation.

The additional benefit is the ease of maintenance on the HP/1000. A procedure using PUGE is all that has to be called, whereby all source files are transmitted to the /3000 and there renamed to suit the file naming conventions of the /3000 (a source &SOURC.JM will become SSOURCJM.HP1000 on the /3000). The whole scratch area then is simply purged.

3.2. PUTSE (PUT SEnsitometric results)

The program pair to transmit measurement data essentially operates in

the same manner as PUGE as far as the communication is concerned. One of the apparent differences is the automatic streaming of batch evaluation jobs by the HP/3000 slave.

The measurements of a day are usually transmitted within about 15 minutes.

3.3. FMDEN (Fetch Master sets for DENsitometry)

By this program pair either a complete dump of all relevant master sets from the /3000 to the /1000 is realised or individual data is updated on both computers simultaneously.

A complete dump takes about one minute and fills 805 sectors of the 7906 disc. This corresponds to a practical speed of 25 kBaud (in the software sense).

4. Design of PTOP program pairs

As a first general rule it can be stated that PTOP communication should be avoided, whenever other means such as remote file access or remote EXEC calls can give the same result, because the application will be operational more rapidly.

There are situations though, as seen earlier, when PTOP communication

is the only good solution for the end user. Even between two HP/3000 such program pairs may be useful as well described in the DS3000 reference manual.

Once it has been decided to use PTOp communication, programing is usually done in such a way that the computer initiating the communication has the master program and the other one the slave. DS3000-1000 appears to have a bug though and it is currently not advisable to write HP/1000 master programs and HP/3000 slaves. Once the programs are in working order there are no big problems to be expected. Testing proves to be very disagreeable though, because an abort of the slave on the /3000 on bounds violation, integer overflow, an inexistent file or other such normalities in a testing phase systematically provokes a system down of MPE. Obviously the poor programmer provoking such a small disaster on a production oriented heavily loaded HP/3000 will have plenty of niceties to hear.

If a /3000 slave has to be written there are a few options which increase the chance of MPE-survival. An obvious solution is to make the slave as short and simple as possible. Delegate more intricate work to previously or later executable programs. One of the possibilities in this context is the use of streamed subsequent jobs. Another solution can consist of a very simple PTOp pair, which only initiates the communication and then is followed by another pair, for which the master is on the /3000 side. In fact it is possible for a slave program itself to become master of another. Probably this would not help to solve the abort problem (it has not been tested), because it appears to be related to the original, atypical scheduling of the slave.

Other obvious means of avoiding the slave to abort consist in all the necessary tests to intercept the potential error. This requires additional code though and thereby makes the testing phase longer. The final way out for testing again is disagreeable: night shift. Wait until all those backlogged batch jobs are out of the system and then learn to warmstart MPE!

There are other potential problems with PTOp. A programmer glancing through the available procedure calls intuitively assumes that communication starts by POPEN and terminates by PCLOSE. The assumption is correct for POPEN, but PCLOSE not only stops communication, but also aborts the slave, even if it is concurrently working for another master. Except for emergency exits a master program will therefore contain at least one POPEN, but usually no PCLOSE.

A slave must be careful by what process it is called. All transactions of the master are received by the GET procedure. By program logic maybe a PREAD is expected to be calling, in some cases it may though be a POPEN from another master or, in fact, from the same master, which terminated and restarted for some reason.

From all that has been said it primarily can be concluded that PTOp communication should be kept as simple as possible, especially if the programmer is inexperienced with DS3000. Hopefully the system down bug will be out before MPE revision Zephyr (or whatever) arrives and then PTOp communication can be a very useful tool even between a /1000 and a /3000.

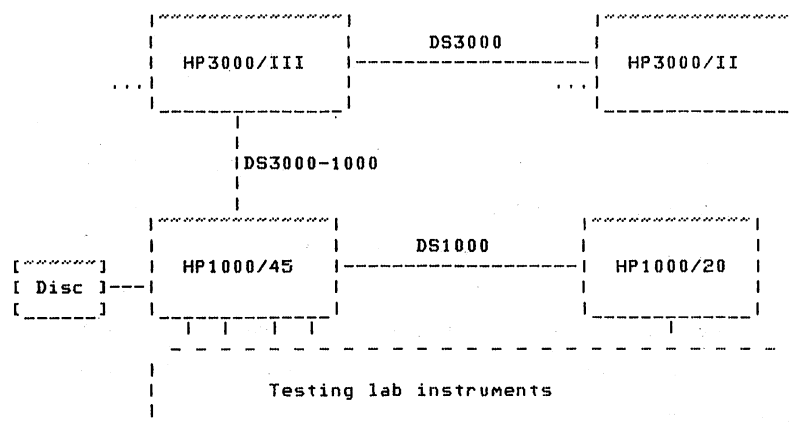


Fig. 1
Configuration with hardwired
connections.