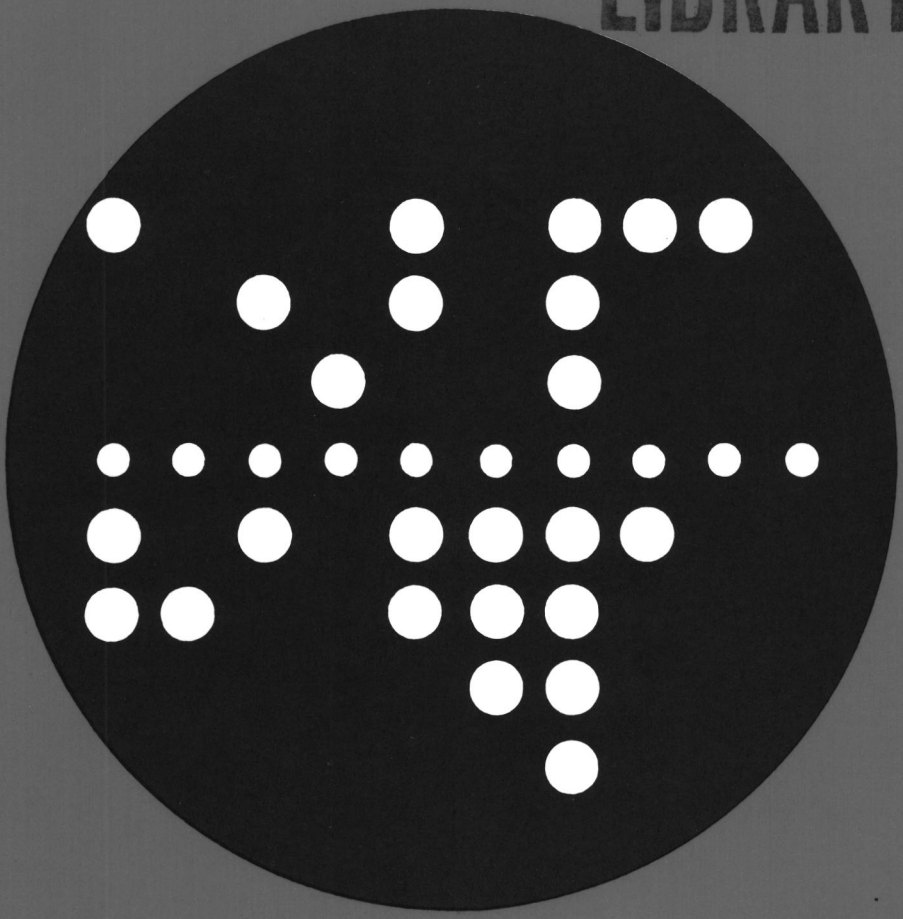
COMPUTING CENTRE NEWSLETTER

November 1981 - N. **56**



Commission of the European Communities

JOINT
RESEARCH
CENTRE

Ispra Establishment

CONTENTS

EDITORIAL NOTE

The Computing Centre Newsletter is published monthly except for August and December.

It describes developments, modifications and specific topics in relation to the use of the computing installations of the Joint Research Centre, Ispra Establishment.

The aim of the Newsletter is to provide information of importance to the users of the computing installations, in a form which is both interesting and readable.

The Newsletter also includes articles which are of intellectual and educational value in order to keep the users informed of new advances in computer science topics.

Administration and contact address:

Ms. A. Cambon (tel. 5446)
Support to Computing
Building 36
J.R.C. Ispra Establishment
21020-ISPRA (Varese)

COMPILATION AND EXECUTION EFFICIENCY

M. Dowell

In the months since the installation of the new AMDAHL 470/V8 system we have been actively checking and analysing the workload profile of the central computer system. One of the main conclusions which has been made concerns the batch usage and in particular the batch scheduling parameters. Apart from the use of specific extra facilities (specified in the $OC records such as: magnetic tapes, paper tape output etc.) there are three main scheduling parameters:

1) Maximum Memory size required (100K, 200K, ...)
2) Type of job (CPU bound or i/o bound)
3) Maximum "time" required (1 min., 2 min., ...)

The first two of these parameters are coded in the same "$ CLASS" record (see Newsletter N. 42), the third parameter is coded in the "$ TIME" record. It is apparent, following the analysis of the present situation, that with the present constraints, the second and third parameters are more important than the first when attempting to make efficient use of the computer resources. In particular, the ability to know that a job will be either CPU bound on i/o bound is extremely important. Many jobs, because of the nature of the various steps which they perform, may be during one particular stage of their execution i/o bound but in another CPU bound.
Many large FORTRAN programs fall into this category. They are compeltely i/o bound during their compilation link edit steps, but CPU bound during their execution. This makes their scheduling extremely difficult.
At present the most severe constraint on our installation is in the handling of input/output transfers. Therefore, the inability to detect that, even through a job may begin by being i/o bound, at a later stage it may have a long period of soley CPU usage, is a considerable problem.
The solution which is suggested in this article to improve this situation involves the splitting of such jobs into a chain of two separate jobs. The first job being a compile and link edit of the program, the second being on execution of the previously compiled loads module version of the library. Both of these jobs may make use of the techniques described in the article "FORTRAN Batch Execution" in this Newsletter. This splitting enables the compile and execution (i/o bound) to be performed in one job which on successful completion "chains-in" the execution job (CPU bound).
It is important to note, however, that although most large FORTRAN programs are CPU bound during their execution, this is not necessarly so.
In many cases large FORTRAN programs may be i/o bound.
The output at the end of the HASP log gives an indication as to whether or not the system regards a particular job as i/o bound or CPU bound.

Example

The example shows the two jobs forming the chain.
Job A, which performs the compilation and link edit of the FORTRAN program (i/o bound requiring 200 K bytes of memory and TIME of 2 minutes).
Job B, which performs the execution of the program from the load module library (CPU bound requires 400 K bytes of memory and TIME of 15 minutes).

A)

```
//          JOB(your JOB card)
$      TIME 002
$      CLASS B
$OC J=CHAIN01,0
//     EXEC FTG1CL
//CMP.SYSIN DD DSN=TSOTEST.PROG1.FORT,DISP=SHR
//LKED.SYSLMOD DD DSN=TSOTEST.LIB.LOAD(TESTPROG),DISP=OLD
```

B)

```
//          JOB(your JOB card)
$      TIME 015
$      CLASS 4
$OC J=CHAIN01,1
//     EXEC XEQH,GOSET='TSOTEST.LIB.LOAD',GO=TESTPROG
//GO.SYSIN DD *
.
.       input data for program
.
/*
```

Note 1
The jobs may be input as decks of cards or SUBMITted from TSO. However, for successful chain activation, **job B must be submitted before job A** (See Newsletter N. 34, "Job Execution Requirements", pages 13 and 14)

Note 2
This type of chaining to separate the CPU bound and I/O bound parts of a job into two separate jobs may be equally applied to other programming language procedures (PL/I, COBOL, PASCAL) when the execution is CPU bound.

The use of this system will be of benefit to both the Computing Centre and the users. It will enable the more efficient scheduling of the system and at the same time users who make use of this system will find that their jobs are being executed more rapidly.
If the two parts are submitted as one job, then they will be seen to be of mixed nature (i.e. part CPU bound, part i/o bound). Because their total execution time is large (15 mins), the running of the job will be delayed longer than would be the two separate well defined type of jobs.

FORTRAN BATCH EXECUTION

M. Dowell

A new job control language procedure has been written which may be used for the execution of a previously compiled FORTRAN program which is held in a load module library. The program is assumed to have been compiled and link edited using either the FORTRAN G1 or the FORTRAN HE compiler and the load module written to a member of a permanent load module library.
For example, the following job would be sufficient to perform this task:

```
//        JOB(your JOB card)
$     CLASS B
//    EXEC FTG1CL
//CMP.SYSIN DD DSN=TSOTEST.PROG1.FORT,DISP=SHR
//LKED.SYSLMOD DD DSN=TSOTEST.LIB.LOAD(TESTPROG),DISP=OLD
```

Where the FORTRAN source program held in TSOTEST.PROG1.FORT and the compiled and link edited load module program is placed in member TESTPROG of load module library TSOTEST.LIB.LOAD.
The program name will be TESTPROG and this must be used in the subsequent procedure to execute the program.

Note
If "DISP=OLD" is used on the LKED.SYSLMOD DD statement, then:
If the member does not exist in the data set it will be created.
If the member already exists then its contents will be overwritten by the new module.

If "DISP=MOD" is used on the LKED.SYSLMOD DD statement, then:
If the member does not exist in the data set it will be created.
If the member already exists it will not be overwritten but an attempt will be made to create a member TEMPNAME. If this also already exists then the link edit will fail.

Similarly for **small programs requiring little compilation and link edit time,** it is possible to perform the compilation and link edit in the TSO foreground region in the following way:

(Note. This fragment of a TSO session is assumed to take place under TSO usedid TSOTEST).

```
fort prog1 (CR)
.
.
.
link (prog1 obj,*) fortlib load(lib load) (CR)
ENTER CONTROL STATEMENTS
name testprog (CR)
(CR)
READY
```

Note the use of the control statement (arising from the * in the first parameter of the LINK command) to define the member name (TESTPROG) to which the load module should be written.
In both cases the result is the setting up of the load module version of the FORTRAN program in the data set TSOTEST.LIB.LOAD(TESTPROG).

The execution of the FORTRAN program in batch may be performed using the new XEQH procedure as follows:

```
//        JOB(your JOB card)
$     CLASS ...
//    EXEC XEQH,GOSET='TSOTEST.LIB.LOAD',GO=TESTPROG
```

Note. Class depends on the memory size and type of program.

The procedure has an internal stepname "GO" and has an allocation for FORTRAN unit 6 (FT06F001) to the lineprinter output spool (SYSOUT=A).
The FORTRAN input unit is assigned DDNAME SYSIN.
Therefore, if the test program reads input on unit 5 and writes "lineprinter" output to unit 6 then the following JCL is necessary.

```
//        JOB(your JOB card)
$     CLASS ...
//    EXEC XEQH,GOSET='TSOTEST.LIB.LOAD',GO=TESTPROG
//GO.SYSIN DD *
.
.
.     data input for program
.
.
/*
```

Note. Class depends on the memory size and type of program.

Similarly the FT06F001 DD statement may be overwritten and/or other DD statements may be added, as necessary, in the following fashion:

```
//        JOB(your JOB card)
$     CLASS ...
//    EXEC XEQH,GOSET='TSOTEST.LIB.LOAD',GO=TESTPROG
//GO.FT06F001 DD DSN=TSOTEST.ABC.LIST,UNIT=DISK,
//            VOL=SER=USER01,
//            DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA),
//            DISP=(NEW,CATLG,CATLG),SPACE=(200,100)
//GO.FT02F001 DD DSN=TSOTEST.DS2.DATA,DISP=OLD
.
.       input data for program
.
/*
```

Note. Class depends on the memory size and type of program.

In the example above the data read on unit 5 (SYSIN) is provided "in-stream". The output on unit 6 is written to a new data set which is created in the step (TSOTEST.ABC.LIST). The output on unit 2 is written to a data set (TSOTEST.DS2.DATA) which already existed before the step.

Note.
It is necessary that all overide DD statements (i.e. the ones for FT06F001) appear before all new DD definitions (SYSIN,FT02F001).

The use of the XEQH procedure enables time to be saved in the case where the same program is run repeatedly with different sets of data. It is also useful when the compile/link and execution are to be split into two jobs which is the situation described in the article "Compilation and Execution Efficiency" in this Newsletter.

§§*§*§*

CONNECTED!

The following extract is not from one of our Newsletters but from the "Bulletin d'information du Centre Interuniversitaire de Calcul de Grenoble" issue number 86 (October 1981). However, the problem under discussion would appear to be of common importance in both centres!

"..... depuis les derniers jours de Juin, les mois de Juillet et Septembre, a part quelques cablages autour de l'autocommutateur, l'essentiel de mon travail a consiste a verifier les lignes existantes, les coupleurs des frontaux, voire même les terminaux chez les utilisateurs. Tout cela grace aux nombreux orages qui se sont abattus sur notre region et qui sont particulierement efficaces sur le campus."

SOFTWARE CHANGES

M. Dowell

During the next few months, a reorganization of the applications software available to users will be performed by members of the Support to Computing group. This reorganization will rationalize the use of the software and make the usage by users more will-defined and systematic. It may, however, require some short-term changes by the users to move into line with the new system. The following information gives the first of a series of changes which will be necessary.

1) **Interactive Graphics Using the PLOT-10 Package**
   Users of the PLOT-10 interactive graphics library who are at present using SYSTSO.TEKLOAD to obtain the relevant routines should now use the library from data set GL01.LIBPLT10. This data set is cataloged. The data set contains the identical set of subroutines to those in SYSTSO.TEKLOAD.
   From the 1st January 1982 the SYSTSO.TEKLOAD library will no longer be available.

2) **Intermediate File to Benson, Gould Plotters**
   Users of these systems must use the procedures (BENSON, GOULD, GOULDX) provided. Some changes have been made within these procedures, which will not, however, affect the user interface.

3) **Use of Mathematical Libraries**
   The data sets containing the subroutines for the NAG and IMSL mathematical libraries (SYS1.LIBNAGS, SYS1.LIBNAGD, SYS1.LIBMASXS, SYS1.LIBMASXD) must always be accessed via the catalog. These data sets will always be accessible via the catalog but the volume on which they reside may change. Details of how to use these libraries via the catalog may be found in the new green book "Using the IMSL & NAG Libraries".

4) **Using the Graphics Library "SYS1.LIBERTY"**
   As with the mathematical libraries, this library must now be used via the catalog. In particular in a batch compilation the "VLB=" and the "ULB=" parameters should no longer be given.
   For example:

```
//         JOB(your job card)
//     EXEC FTG1CG,PRN=ERTY
//CMP.SYSIN DD *
.
.   FORTRAN program using the graphics library
.
/*
```

LES SYSTEMES CONVERSATIONNELS

J. Pire

L'ordinateur du C.C.R. supporte deux systèmes conversationnels T.S.O. (Time Sharing Option) et I.M.S. (Information Management System.

La conception à la base de ces deux systèmes est cependant complètement différente.

Alors que T.S.O. est surtout basé sur l'exécution de commandes ou de programmes mais veille à une répartition "équitable" des ressources entre les différents usagers, I.M.S. laisse aux analystes et programmeurs le soin de répartir ces ressources de façon raisonable.

Lorsqu'une ligne de communication est partagée entre différents utilisateurs, T.S.O. veille à ce que chaque utilisateur n'envoie qu'un nombre limité de caractères à la fois et n'accapare pas pendant un temps trop long la ressource "ligne de communication" au détriment d'autres utilisateurs dépendent de la même ligne.

I.M.S. ne donne pas cette guarantie. Il en résulte que si l'analyste ou le programmeur qui ont conçu ou développé l'application ont défini des messages trop longs, la ligne de communication est monopolisée pendant plusieurs secondes au détriment des autres utilisateurs qui ont la sensation que leurs terminaux ne fonctionnent plus où que le système est perdu.

Voyons quels sont les conditions qui peuvent provoquer un tel état de chose.

Les lignes de télécommunications qui connectent les unités de contrôle des terminaux vidéo à l'ordinateur (ou plus exactement aux "front-end" ITT) peuvent transporter chacune 4800 b.p.s. (bit par seconde). Un caractère est composé de 8 bits par conséquent le traffic maximum est de 600 caractères par seconde.

Un écran compte environ 1900 caractères. La transcription d'un écran complet monopolise déjà la ligne de communication pendant plus de 3 secondes mais les implementations de certaines applications I.M.S., compte tenu de certains caractères de contrôle et de la façon dont l'application à été conçue, envoyent parfois beaucoup plus de 2000 caractères en une seule fois ce qui peut monopoliser la ligne pendant 5 à 6 secondes.

Une seconde remarque doit être faite. Le niveau moyen d'erreur sur les lignes de communication est de l'ordre de 10^{-4} à 10^{-6}. Plus les messages sont longs plus elevée est la probabilité d'une erreur pendant la transmission et plus elevée est la probabilité d'une retransmission du message.

Un message de 4 secondes qui doit être répété monopolise la ligne pendant plus de 8 secondes s'il est répété 1 fois et 12 secondes évidemment d'il doit être répété deux fois.

Une message demandant 4 secondes de transmision comporte 2400 caractères soit 19200 bits.
Nous avons relevé des messages de 3700 caractères soit 29600 bits nécessitant 7,65 secondes de transmission.
Nous vous laissons imaginer la sensation qu'éprouve un utilisateur de T.S.O. qui reçoit des messages qui ne dépassent généralment pas 80 caractères (640 bits ou 0,15 sec.) lorsqu'ils constatent que leur terminal à certain moments restent bloqués pendant plus de 10 secondes.

Conclusion.

Si vous utilisez un terminal vidéo attaché à un controleur qui sert également d'autres utilisateurs employant I.M.S., ne vous énervez pas, ce n'est généralement pas eux qui ont développé l'application; ils doivent eux aussi travailler et n'ont probablement pas le choix d'un autre terminal.

Si vous êtes utilisateur T.S.O. cherchez un autre terminal: le Centre de Calcul et le matériel ne sont pas responsables de votre mésaventure.

ARRIVEDERCI

Mr. Dieter König, head of the Systems Software Sector of the Computing Centre, is leaving the Centre at the end of November 1981. He will take up an eqvivalent post in the Computing Centre at the E.E.C. - Luxembourg.
We would like to take this opportunity to recall his excellent work in the Computing Centre over the last 5 years. Among his many achievements, special mention should be made of his work, with his colleagues in the System Software Sector, in the implementation of the TSO system and more recently the specification of the software requirements for the call for tender and the subsequent implementation of the basic software for the AMDAHL computer system.
We wish him great success and happiness in his new post.

BENVENUTO

Mr. Hervé de Sadeleer has recently joined the Informatics Support Sector.
His work in the sector will be concentrated in the areas of the Program Distribution Agency and "large-scale" computing. He will also be involved in the direct support consultancy service, initially covering FORTRAN and TSO problems.
He was formerly the informatician in the Atmospheric Pollution Sector of the Belgian Royal Institute of Meteorology. In his work he was involved in the setting up of the National Data Processing Centre Network and the provision of informatics support to the group involved in atmospheric pollution

---

Mr. Jacques Locquet has recently joined the Teleinformatics Sector of the Computing Centre.
He is charged with the design of a satellite data acquisition facility which is to be integrated with our currently available terrestrial networks. In the framework of the COST 11 bis international research project he is collaborating with many other research laboratories with the aim of exploiting the Orbital Test Satellite for high speed data transmission over continental distances.
He was formerly a research assistant in Computer Science at the "Faculté Polytechnique de Mons" in Belgium.

STATISTICS OF COMPUTING INSTALLATION UTILIZATION.
REPORT OF COMPUTING INSTALLATION EXPLOITATION
FOR THE MONTH OF OCTOBER 1981.

	YEAR 1980	YEAR 1981
General		
Number of working days	23 d	22 d
Work hours from 8.00 to 24.00 for	16.00h	16.00h
Duration of scheduled maintenance	22.67h	12.17h
Duration of unexpected maintenance	18.17h	17.83h**
Total maintenance time	40.84h	30.00h
Total exploitation time	327.16h	331.00h++
CPU time in problem mode	320.72h+	444.26h*
Batch Processing		
Number of jobs	9005	9217
Number of cards input	798000	205532
Number of lines printed	28577000	24642400
Number of cards punched	37000	8850
CPU Time	288.00h+	370.03h*
Number of I/O (Disks)	24304000	25760900
Number of I/O (Magnetic tape)	4872000	8185900
T.S.O.		
Number of LOGON's	5164	5716
Number of messages sent by terminals	333347	369346
Number of messages received by terminals	2243198	2506760
CPU time	29.78h+	65.49h*
Number of I/O (Disk)	4513000	3948240
Connect time	3002.90h	4260.02h
ADABAS		
Total time service is available	-	165.01h
CPU time	-	3.70h*
Number of I/O (Disk)	-	784500
IMS		
Total time service is available	133.55h	116.25h
CPU time	2.94h+	5.04h*
Number of I/O (Disk)	896000	856780

\+ Real CPU has been multiplied by a factor of 2.0 to indicate the increased throughput of the AMDAHL 470/V7A.

++ Including 9.00hrs overtime.

* Real CPU has been multiplied by a factor of 2.5 to indicate the increased throughput of the AMDAHL 470/V8, in comparison with the IBM 370/165.

** Covering all the configuration.

UTILIZATION OF COMPUTING CENTRE BY OBJECTIVES & APPROPRIATION ACCOUNTS FOR THE MONTH OF OCTOBER 1981.

		AMDAHL 470/V8 work units in hours
33001	Reactor Safety	347.94
33002	Plutonium Fuel and Actinide Research	-
33003	Safety of Nulcear Materials	4.71
33004	Fissile Materials Control and Management	13.72
33005	Super-SARA Test Programme SSTP	191.52
33011	Solar Energy	1.10
33012	Hydrogen Production, Energy Storage and Transport	2.14
33013	Thermonuclear Fusion Technology	31.88
33014	High Temperature Materials	2.67
33021	Protection of the Environment	23.66
33022	Remote Sensing from Space	3.12
33041	Informatics	73.59
33043	Support to the Community - Bureau of References	0.01
33044	Training and Education	-
33046	Provision of Scientific and Technical Services	13.06
1.20.1 1.20.2	General Administration - JRC General Services - Administration - Ispra	81.14
1.20.3	General Services - Technical - Ispra	0.38
1.30.0	Central Workshop Ispra	1.70
1.40.2	ESSOR	8.45
	TOTAL	800.79
1.94.0	Services to External Users	5.22
	TOTAL	806.01

BATCH PROCESSING DISTRIBUTED BY REQUESTED CORE MEMORY SIZE

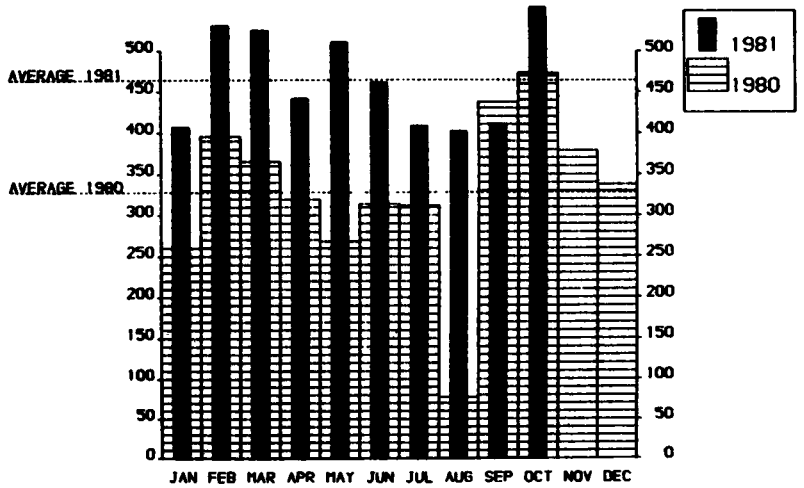
	100 k	200 k	300 k	400 k	600 k	800 k	1000 k	1200 k	1400 k	1400 > k
No. of jobs	2773	1716	1352	972	1088	287	145	25	18	222
Elapsed time	78	145	271	206	174	115	93	36	4	85
CPU time	5.3	26.1	83.2	46.0	47.0	44.7	52.2	14.9	2.3	47.8
"Equiv" time	23	46	132	82	73	61	57	21	3	55
"Turn" time	0.5	2.4	4.0	4.2	4.1	5.5	5.4	6.2	4.2	5.5
I/O (disk)	1539	2636	5766	4932	2943	2259	677	927	54	1089
I/O tape	2133	561	2933	591	1694	109	101	3	-	1

NOTE.
All times are in hours.
"Equiv" means equivalent.
"Turn" means turn around.
All I/O transfers are measured in 1000's.

PERCENTAGE OF JOBS FINISHED IN LESS THAN:

TIME	15mn	30mn	1hr	2hrs	4hrs	8hrs	1day	2day	3day	6day
%year 1980	40	56	69	82	92	97	99	100	100	100
%year 1981	30	42	53	66	80	94	99	100	100	100

HISTOGRAM OF TOTAL EQUIVALENT TIME(HRS)



Projected total for 1981 = 5574 Hours(using average)

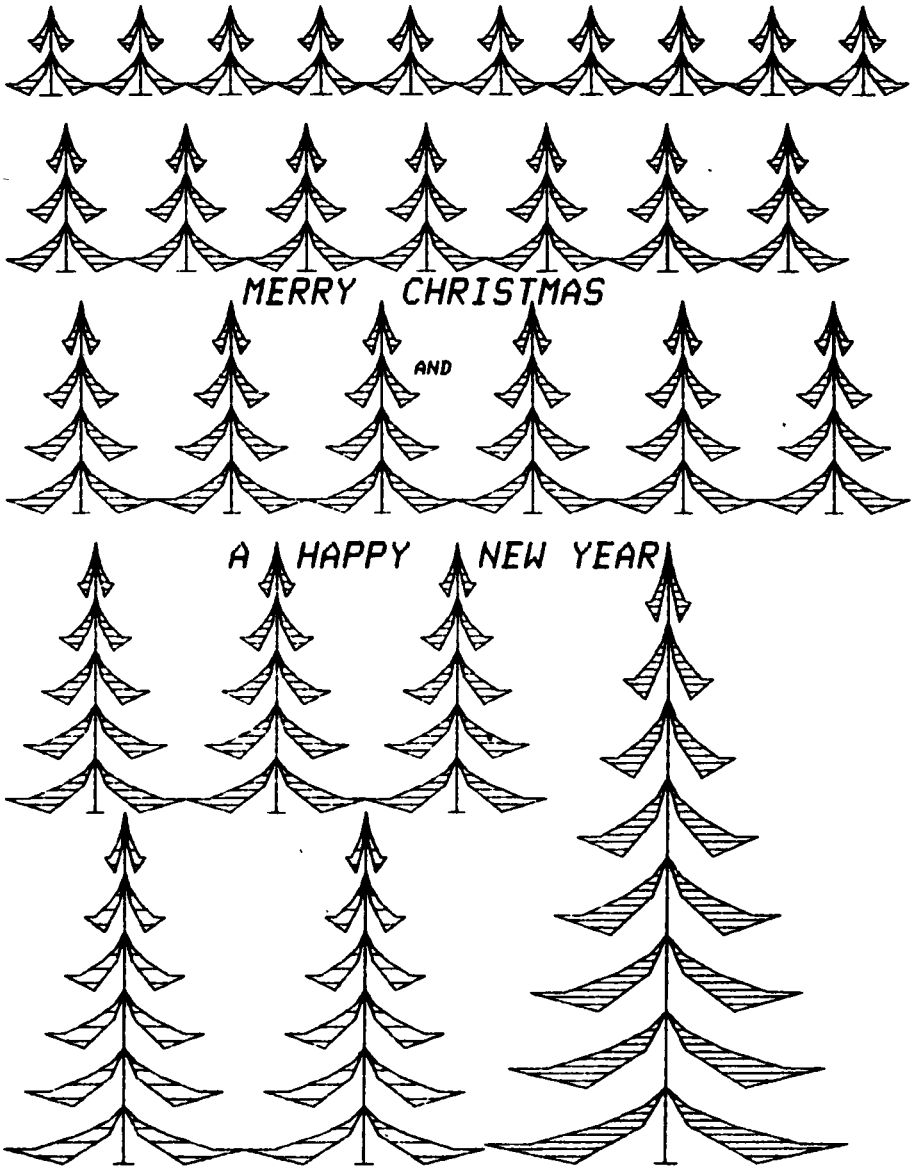
Total for 1980 was = 3936 Hours

REFERENCES TO THE PERSONNEL/FUNCTIONS OF THE COMPUTING CENTRE

			Room	Tel.
Manager of the Computing Centre		J. Pire		
Responsible for User Registration		Ms. G. Rambs		
Operations Sector				
Responsible for the Computer Room		A. Binda-Rossetti		
Substituted in case of absence by:				
Responsible for Peripherals		G. Nocera		
Systems Software Sector				
Responsible for the sector		D. König		
Substituted in case of abscence by:		P.A. Moinil		
Responsible for TSO Registration		C. Daolio		
Informatics Support Sector				
Responsible for the Sector	(f.f.)	H. de Wolde	1883	5787
Secretary		Ms. G. Hudry	1873	5787
Responsible for User Support		M. Dowell	1886	5419
General Inf./Support Library		Ms. A. Cambon	1871	5446
Advisory Service /List of Consultants(See Note 1)			1870	5446

A. Inzaghi — H. I. de Wolde
A. A. Pollicini
R. Meelhuysen — M. Dowell

Note 1. The advisory service is available in the same room as the Computing Support Library (room 1870). Exact details of the advisory service times for a specific week can be found at the head of any output listing (for that week).
Any informatics problem may be raised. However, the service is not designed to help users with problems which are their sole responsability. For example, debugging of the logic of programs and requests for information which can easily be retrieved from available documentation.
If necessary, other competent personnel from the informatics division may be contacted by the consultant but not directly by the users.
The users should only contact the person who is the consultant for that specific day and only during the specific hours.
Outside the specific hours general information may be requested from Ms. A. Cambon in the Computing Support Library.


MERRY CHRISTMAS
AND
A HAPPY NEW YEAR

HOW TO OBTAIN COMPUTING CENTRE DOCUMENTATION

Person interested in receiving copies of the Computing Centre "green books" or in receiving regularly the "Computing Centre Newsletter" are requested to complete the appropriate part of the following form and send it to:

Ms. A. Cambon

Support to Computing

Building 36

Tel. 5446.

Indicate with a (√) which option are required.

Please add my name to Newsletter mailing list ( )

Please send me copies of the following "green books":

JRC-TSO Primer ( )

JRC Computer Graphics (new version) ( )

Towards a New Programming Style ( )

LIBRARIAN ( )

Using the IMSL & NAG Libraries ( )

NAME .....

ADDRESS .....

.....

.....

TELEPHONE .....