Calhoun: The NPS Institutional Archive

DSpace Repository

Information Technology and Communication Services (ITACS)Computer Facility Newsletter, 1968-1999

1973-11-21

W.R. Church Computer Center Newsletter / 1973-11-21

Monterey, California, Naval Postgraduate School

https://hdl.handle.net/10945/57297

W. R. CHURCH

Computer Center

NEWSLETTER

November 21, 1973
Volume 5, Number 6

TABLE OF CONTENTS

STAFF PROGRAMMING ASSISTANCE

One of the duties of the professional programming staff is to help individual students and faculty members program their applications on the IBM 360/67. Usually this cooperation has to be limited to consulting help (e.g., on computational methods, JCL, etc.) but on occasions can extend to substantial programming assistance when there is a clear benefit to a group of users rather than an individual. Examples of such projects include joint development of some innovative use of computers in classwork or research (e.g., use of interactive graphics to simulate an experiment); creation of a new computer-based service or functional capability; filling of a gap in the existing program and subroutine libraries; programming support for mini-computer systems connected to the IBM 360/67.

We invite users, faculty members and students, to help us identify potential projects in these or other areas. We have a lot of ideas, but you may have better ones or, by your input, cause us to revise present priorities.

The manpower resources available for such programming projects are limited and fluctuate with time. However, small projects can often be interleaved with larger assignments--we try to multiprogram the staff as well as the computer system! Obviously, we want to apply available programming time to maximum advantage to the School. When there is more demand for programming assistance than can be scheduled, projects are chosen according to their potential usefulness to numbers of users. Higher priority is given to development projects which open up new areas of computer applications of benefit to classwork and/or research at NPS.

Users who have ideas or need programming help are asked to fill out a short form, copies of which are available in the Information Services Office, In-147 or from the User Services Manager, In-133. The form is designed to gather sufficient background information on potential projects.

If you would like or need our help in formulating your idea please come and discuss it with Roger Hilleary, Manager, User Services (In-133, X2752).

ECOLOGICAL NOTES

1. The Computer Center is joining with FNWC and EPRF in the recycling of computer output paper and punched cards. When the details are worked out with the receiver in San Jose, we hope to have special receptacles available to collect these items.

However, you don't have to wait. Please bring your old punched cards to the Keypunch Room, In-141 and give them to one of the operators. Printer paper will be accepted in the Output Room, In-140. Every little bit helps. You can prevent valuable (and scarce) paper from becoming landfill and possibly save a tree from being felled.

2. Please remember to turn off the keypunch when you are finished and remove all your used cards. The first act saves electricity; the second improves the environment for the next user, your colleague.

FORTRAN FORMAT HINT

Fortran programmers often seem to have problems with FORMAT statements when they try to use a repeat count on a multi-line output. For example, let us assume the integers 0(1)9 are stored in array M(10) and we desire to print them two on a line, double spaced. We might write

```
          WRITE (6,1) M
     1    FORMAT ('0',2(I1, 1X))
```

and expect to achieve our purpose. However, what we get looks like this:

```
          0 1
            3
            5
            7
            9
```

(where integers 2, 4, 6, and 8 have disappeared into Column 1). This
seems to be caused by a Fortran quirk which results in the '0' control
character not to be scanned as expected.

However, there is a simple remedy to this problem which has long
been available to the cognoscenti. Simply write the FORMAT statement:

```
1       FORMAT (('0',2(I1,1X)))
```

i.e., with redundant parentheses. This forces repetition of double-
spacing at exit from the inner parenthesis. It will provide the re-
quired (double-spaced):

```
          0 1
          2 3
          4 5
          6 7
          8 9
```

THE COMPUTER CAN'T READ SWAHILI

Several times a month the Computer Center staff is confronted with
trying to make our 360/67 read and understand a reel of tape written by
another machine. The frustrated owner of the tape usually wants to
begin immediately to use the data on the tape and not spend a week just
getting our computer to read the tape. A little foresight in ordering
the tape can save the user many hours of uninteresting work.

First, if the originating installation has 9-track tape drives, by
all means, order a 9-track tape. If the originating installation uses
IBM equipment and IBM's standard OS tape labels, request such labels. If
you do, all you will have to know to use the tape is the name of the
reel, and the DSNAME of the files you want. If the originating installa-
tion is non-IBM or has its own in-house tape labels, request a non-labeled
tape and specific information about the record lengths and blocking factors
on the tape. CAUTION: If you order a 9-track tape, make sure the record-
ing density is 800 BPI (bits per inch). We CAN NOT read 1600 BPI and any
such tape will have to be re-ordered.

Next, if the originating installation has only 7-track tapes, re-
quest that the tape be "BCD" and "even parity". Although our computer can
read 200, 556, and 800 BPI densities, experience has shown that 556 BPI
is more reliable than 800 for a 7-track tape. As in the 9-track case, if
IBM tape labels are available, request them. If not, request a non-labeled
tape and specific information about record and block sizes.

In all cases, a partial listing of the contents of the tape produced at the originating installation will be invaluable in determining if our computer is really reading what is on the tape.

Finally, we realize that there will have to be many exceptions to the above recommendations due to unavailable features or incompatabilities at the originating installation. If this is true, our duty consultant will be happy to advise you as to the most-nearly compatible tape format to order. Also, Technical Note 0211-08 includes much useful information on this subject.

FORTRAN EXPONENTIATION

(The following item originally appeared in the MIT Information Processing Center _Bulletin_).

For small integer exponents, exponentiation will generally be performed faster if the exponent is expressed as an integer constant or variable, rather than as a real constant or variable. This is because different algorithms are used for integer and real exponentiation.

A number raised to a real power, A**B, is evaluated as EXP (B*ALOG(A)). This involves both the exponential and logarithmic subroutines and takes approximately as long as twenty multiplications, regardless of the magnitude of B. If A is negative, an IHC253I error message will be produced, the sign of A will be changed, and execution will continue.

The algorithm for raising a number A to an integer power N is as follows: first by successive squaring evaluate A^2, A^4, A^8, A^{16}, etc., up to and including the largest exponent that is smaller than or equal to N; then, multiply together all those squares whose exponents add up to N. For example, to evaluate A**83, compute A^2, A^4, A^8, A^{16}, A^{32}, A^{64}; then multiply $A*A^2*A^{16}*A^{64} = A^{(1+2+16+64)} = A^{83}$. This involves nine multiplications. The largest possible integer exponent, $2^{31} - 1$, also involves the greatest number of multiplications, sixty. Raising a number to this integer power will take three times as long as raising a number to the same power expressed as a real number.

For integer values of B between 2,047 and 1,048,576 there should be no significant timing differences between the two methods. However, always using the form A**N is probably a good rule of thumb.

PROGRAMMING AIDS LIBRARY

Technical Note 0211-25 on the Programming Aids Library has been revised. Programs EDIT and FTIDY have been added. EDIT is an all-CAPITALS text editing program which was mentioned in the last Newsletter. FTIDY is a program that re-formats and re-numbers Fortran source decks that are difficult to read due to patches and revisions. It also re-sequences statement numbers and cards.

The program NEWDEK has been revised to translate and punch a source deck from EBCDIC to BCD as well as vice versa. This option has been added to allow easier "transportation" of source programs to non-IBM computer systems.

REVISED TECHNICAL NOTES - CP/CMS AND COBOL

Another two of the Center's Technical Notes have recently been revised and reprinted for the needs of our users. TN-0211-11 is a COBOL User's Guide by L. M. Brunner and C. W. Kellogg. It supersedes two technical memoranda on ANS COBOL and WATBOL, respectively. TN-0211-24, edited by K. B. Strutynski, is a CP/CMS User's Guide. Copies of this guide have been placed at the IBM 2741 terminals. Individual copies of either technical note are available in In-147.

IMSL LIBRARY - LOAD MODULE LIBRARIES

Precompiled modules of both the single-precision and double-precision routines of Edition 3 of the International Mathematical Statistical Library are now available. Included in these two load module libraries are both the new abilities added by Edition 3 and those Edition 2 abilities that were sufficiently modified to warrant recompilation, plus those routines from Edition 2 that were left unchanged.

Use of these libraries should be a considerable convenience for any user who desires to call an IMSL routine without modification, since it eliminates having to determine and reference all the service sub-modules used by the desired routine.

To access the single precision library, your OS deck (not QUICKRUN) should look like:

```
//          (Standard OS JOB card - see Section 3.3.2.1 of User's Manual)
//   EXEC   FORTCLG (etc. as required)
//FORT.SYSIN  DD  *
            (Your FORTRAN Source Deck)
/*
//LINK.SYSLIB  DD
//  DD
//  DD  DSN=SYS3.IMSL.SP,DISP=SHR                              #
//GO.SYSIN  DD  *
            (Your Data Deck)   optional
/*
```

Notes: 1) Repeating information in operand fields is not necessary when concatenating an additional library.

2) To access the double precision library, replace the card flagged with # above by:

```
//  DD  DSN=SYS3.IMSL.DP,DISP=SHR
```

a) Utilization by Programming System: (% of total number of jobs, % of total CPU time)

Programming System	All Jobs	QUICKRUN Only
FORTRAN G	(44.4, 30.8)	(48.3, 84.9)
WATFOR G	(16.8, 0.3)	(40.5, 12.3)
FORTRAN H	(2.3, 36.2)	-
XPL	(2.8, 1.0)	-
CSMP	(2.0, 1.0)	-
DSL	(3.5, 7.2)	-
PL/1	(1.1, 0.7)	-
ASSEMBLY	(0.7, 0.3)	-
SMIS	-	(2.7, 0.2)
ALGOLX	(0.4, 0.0)	(1.1, 0.7)

b) Turn Around Time (Min.) by Job Class

Job Class	A	B	C	D	E	F	G	H	I	J	K	O	L	M	Q	ALL	NON-Q
Percentage of Total	8.0	11.6	7.6	9.1	7.5	1.3	2.4	2.1	0.7	2.3	0.5	0.2	2.2	3.1	41.4	100.0	58.6
Turn Around Time (Min.) Average	28	10	14	17	28	37	40	37	41	102	459	78	27	232	5	25	39
Median	4	6	7	9	12	6	12	13	21	33	270	57	9	125	5	6	9

c) Turn Around Time (Min.) by Core Size

CORE SIZE	NUMBER OF JOBS	AVERAGE	MEDIAN
0 to 100K	16676	14	5
101 to 150K	3181	29	10
151 to 250K	918	85	22
251 to 350K	505	218	70
351 to 450K	96	181	100
451 to 550K	2	323	323
551 to ***K	8	633	612

RECENT ADDITIONS TO COMPUTER CENTER LIBRARY (In-162)

Books

Author	Title
Rustin	Algorithm Specification
Hoare	Operating Systems Techniques
Furman	Use of Computers in Engineering Design
Bohl	Introduction to PL/I Programming and PL/C
Rosenfeld	Picture Processing by Computer
Ortega	Numerical Analysis
Stewart	Introduction to Matrix Computations
Gotlieb & Borodin	Social Issues in Computing

Proceedings

Author	Title
Naval Ship Research and Development Center	ADP Data Security and Privacy: Conference on Secure Data Sharing
Southern Regional Education Board and Georgia Institute of Technology	Proceedings of Conferences on Computers and Undergraduate Curricula, 1970, 1971, 1972, and 1973
Share XL	Proceedings of Share XL, Vols. 1 and 2
BCS	International Conference on Computers in Architecture

Reports

Number	Title	Author
1044	Design Trade-offs – Software Associative Memory	Berkowitz
1046	Soviet Cybernetics Review	Holland
1047	Lanczos Algorithm in Numerical Linear Algebra	Golub
1048	Militran	ONR
1049	ORNL Computer Network	Winkler, et al.
1052	Symbolic Algebraic Languages – An Introduction	Morris
1053	Engineering Software Coordination (Vol 1)	Colo. University
1054	Computers in Undergraduate Curricula	New York State U.
1055	Trends in Hardware Technology – in the 1980's	Turn
1056	Design of an Emulator for Computer Systems Research	Iliffe

Defense Documentation Center (DDC)

AD Number	Title	Author/Organization
763359	Scientific Analog Data Processing – Naval Underwater Systems Center	Quigley
763748	Natural Communication with Computers IV	Beranek & Newman
763230	File Organization for Minicomputer Information Retrieval	GTE Sylvania, Inc.
763498	Compiler Design for the Illiac 4	Millestein
763601	Ordered Hash Tables	Amble, et al.
763573	Parallelism in Fast Fourier Transform Hardware	Gold
760006	Complexity of Sequential and Parallel Numerical Algorithms	Carnegie-Mellon U.
760143	Scheduling Parallel Processable Tasks for a Uniprocessor	Fox, et al.
718406	Privacy and Security in Banks	Garrison, et al.
212615	Comprehensive Computer-Aided Building Design System	M.I.T.

AD Number	Title	Author/Organization
731310	Artificial Intelligence	AGARD
762513	PMSL, Interactive Language – System-Level Description and Analysis Structure	Carnegie-Mellon U.
760298	Intelligent Satellites for Interactive Graphics	Van Dam, et al.
759267	Data Base Task Group Report: Illustrative Example	Frank, et al.
210908	Study of Position Titles in the Computer Systems Field	Kuch
862954	Display Evaluation Programming	Mc Grath, Kling
762514	HYDRA: Kernel of a Multiprocessor Operating System	Wulf
762524	Techniques for Analyzing Memory Interference in Multiprocessor Systems	Bhandarker, et al.
762774	Measurement Data – Working Set Replacement Algorithm and Applications	Oliver, et al.
762578	Exploratory Experience with Easy Computer Systems	Sackman
762597	System Modeling and Statistical Data Processing	Kailath, et al.
761604	Surface Modeling and Analysis Subroutine Package	Rosenshine
761969	Three-Dimensional Input for Computer Graphics	Schadel, et al.
761641	Hierarchical Robot Planning and Execution System	Stanford Research Institute
761610	Feasibility of Emulating AN/UYK-7 Computer on AADC Signal Processing Element	Rauscher
762017	Visible Surface Algorithms for Quadric Patches	Mahl
762023	Hidden Surface Line Drawing Algorithm	Archuleta
762018	Computer Display of Curved Surfaces	Gouraud
761563	Privacy Transformations for Data Bank Systems	Rand Corp.
762004	Real Time Visible Surface Algorithm	Watkins
760001	Storage Hierarchy Systems	Madnick
760496	Olive-Drab Minicomputer	Schwenk
761175	Performance of an I/O Channel with Multiple Paging Drums	Fuller