

A. Brown
WS/G-RBK/

NATIONAL RADIO ASTRONOMY OBSERVATORY

Charlottesville, Virginia

Computer Division

June 1, 1971

TO: All summer students, all scientific staff, all division heads,
all computer division personnel

FROM: G. Conant *GC*

SUBJECT: Summer student usage of IBM 360 computer

1. Users of the IBM 360 computer, when submitting a card deck for a computer run, should always attach a white Job Submittal card. These are filled in by hand (not keypunched). Blank Job Submittal cards are found on the job-submittal shelf.
2. One of the items required on the Job Submittal card is the job-class code, which is one of the letters B,C,D,E,F,L,O,N, and which designates a job-priority according to criteria of expected running time, size of memory-partition, etc. This written code should correspond to the class parameter keypunched into the first (JOB) card of the card deck. Example: //name JOB,CLASS=B.
3. This year all summer students are required to prefix the written job-class code (on the job submittal card only) by the letter S. The punched class parameter (on the first card of the card deck) is not prefixed thus, but follows the old pattern. Example: if the keypunched JOB-card contains the parameter CLASS=D, a summer student submitting this card deck for a computer run must write SD for job class on the accompanying Job Submittal card.
4. This written indication of whether a computer job is being submitted by a summer student will be used by the computer operators, to choose between student and non-student jobs when backlogs occur. In those cases, summer students will be given lower priority. Most of the time there should be no noticeable difference.
5. Students who are debugging new programs, running test cases, etc., are urged to keep their computer runs and their printed output as short as possible. If many long jobs are submitted, the computer will become overloaded and a backlog of waiting jobs will develop. Under the above rules of priority, all summer student jobs will be the first to suffer under those conditions.

NRAO JOB CODE (Revised 05/27/71)

S = D: Debugging stage of a new program.
P: Production stage of a program.
C: Change stage of a program.

t = 1: No NRAO telescope involved.
2: Future NRAO telescope involved.
3: More than one NRAO telescope involved.
4: 300' telescope.
5: 140' telescope.
6: Interferometer.
7: 36' telescope (Kitt Peak).
8: Little Big Horn,
9: 85'/1 telescope. }
10: 85'/2 telescope. } if used for single dish observations
11: 85'/3 telescope. } not related to interferometer work.
12: 42' telescope.

r = 1: No receiver involved.
2: Future receiver involved.
3: More than one receiver type involved.
4: 413 channel autocorrelation receiver - "Model II"
5: Multichannel filter receiver (40/50 channel back-ends).
6: Single continuum receiver.
7: Multifrequency (or multifeed) continuum receiver system.
8: Interferometer, continuum correlation system.
9: VLB terminal
10: Interference measuring equipment.
11: Systems to receive meteorological data.
12: 100 channel autocorrelation receiver - "Model I"
13: 384-channel autocorrelation receiver - "Model III"

d = 1: Simulation of observations.
2: Preparation of observations, Ephemeris work.
3: First stage of reduction of observed data.
4: All following stages of reduction of observed data.
5: Analysis stage of an observing program
6: Presentation of observed data.
7: General theoretical work.
8: Development of NRAO Library subprograms.
9: Study, design, and test of equipment.
10: Tape copying,
11: Investigation of errors on magnetic tapes.
12: Listing of card decks.
13: Reproducing of card decks.
14: Assembly and simulation of on-line computer programs.
15: Miscellaneous.
16: Systems development and maintenance. }
17: Systems tasks. } Computer Division
18: Computer accounting.
19: Tape accounting.
20: Electronics division.
21: Administration, Fiscal, and Library.

This is a brief summary showing headlines only. More detailed explanations will be found in the memorandum of 05/16/69 (revised).

Job class codes

Jobs submitted under the following five classes must fit into 90K bytes of storage, and may use any combination of one seven, and two nine track tape drives. Time estimates are given in terms of CPU time rather than clock time.

- B less than 2 minutes
- C 2 - 5 minutes
- D 5 - 10 minutes
- E 10 - 20 minutes
- F over 20 minutes

All other jobs should fall into one of the following classes.

- L jobs requiring up to 180K bytes of storage and the same tape requirements as above.
- N Plotting jobs.
- O Jobs requiring special scheduling. This includes all jobs requiring over 180K bytes of storage, more tape drives than allowed above, or any other special considerations before scheduling. These jobs will have a lower priority than the classes above. Please indicate the storage requirements on the job submittal card for all O class jobs.

NPAO cataloged procedures and examples of their use

FORTC Compile a FORTRAN source deck using the FORTRAN (G) level compiler
FORTGC

```
//jobname JOB (u,s,t,r,d),programmer,CLASS=class  
//     EXEC FORTGC  
   --FORTRAN source decks--
```

FORTCL Compile FORTRAN source decks and then link edit them with
FORTGCL possible object decks.

```
//jobname JOB (u,s,t,r,d),programmer,CLASS=class  
//     EXEC FORTCL  
//FORT.SYSIN DD *  
   --FORTRAN source deck--  
//LKED.SYSIN DD *   if object decks are to be included  
   --object decks--
```

FORTCLG Compile FORTRAN source decks, link them with possible
FORTGCLG object decks and then execute the resulting load module.

```
//jobname JOB (u,s,t,r,d),programmer,CLASS=class  
//     EXEC FORTCLG  
//FORT.SYSIN DD *  
   --FORTRAN source decks--  
//LKED.SYSIN DD *   if object decks are to be included  
   --object decks--  
//GO.SYSIN DD *  
   --input data--
```

FORTLG Link edit a series of object decks and then execute the
 resulting load module.

```
//jobname JOB (u,s,t,r,d),programmer,CLASS=class  
//     EXEC FORTLG  
//LKED.SYSIN DD *  
   --object decks--  
//GO.SYSIN DD *   if input data is to be included  
   --input data--
```

FORTHCL Each of these procedures acts in the same way as the
FORTHCLG corresponding FORTRAN (G) procedure except that they call
FORTHCLG in the FORTRAN (H) level compiler. Because the FORTRAN (H)
 compiler requires more storage, CLASS=L must be specified on the
 job card.

PL1C Compile a source deck written in the PL/1 language
PL1LFC

```
//jobname JOB (u,s,t,r,d),programmer,CLASS=class  
//     EXEC PL1C  
//PL1L.SYSIN DD *  
    --PL/1 source deck--
```

PL1CL Compile a PL/1 source deck and then link edit it allong
PL1LFC with possible object decks.

```
//jobname JOB (u,s,t,r,d),programmer,CLASS=class  
//     EXEC PL1LFC  
//PL1L.SYSIN DD *  
    --PL/1 source decks--  
//LKED.SYSIN DD *     if object decks are included  
    --object decks--  
//
```

PL1CLG Compile a PL/1 source deck, link it with object decks and
PL1LFC then execute the resulting load module.

```
//jobname JOB (u,s,t,r,d),programmer,CLASS=class  
//     EXEC PL1LFC  
//PL1L.SYSIN DD *  
    --PL/1 source decks--  
//LKED.SYSIN DD *     if object decks are to be included  
    --object decks--  
//GO.SYSIN DD *  
    --input data--
```

PL1LG Link edit and execute PL/1 object decks.
PL1LFC

```
//jobname JOB (u,s,t,r,d),programmer,CLASS=class  
//     EXEC PL1LG  
//LKED.SYSIN DD *  
    --object decks--  
//GO.SYSIN DD *  
    --input data--
```

ASMFC Compile source code written in the 360 F level ASSEMBLER language.

```
//jobname JOB (u,s,t,r,d),programmer,CLASS=class
// EXEC ASMFC
//ASM.SYSIN DD *
--ASSEMBLER source deck--
```

ASMFCCL Assemble and link edit

```
//jobname JOB (u,s,t,r,d),programmer,CLASS=class
// EXEC ASMFCCL
//ASM.SYSIN DD *
--ASSEMBLER source deck--
//LKED.SYSIN DD 8 if objects are to be included
--object decks--
```

ASMFCCLG Assemble, link edit and then execute assembly language

```
//jobname JOB (u,s,t,r,d),programmer,CLASS=class
// EXEC ASMFCCLG
//ASM.SYSIN DD *
--ASSEMBLER source deck--
//LKED.SYSIN DD *
--object decks--
//GO.SYSIN DD *
--input data--
```