ONLINE Compilation with a new compiler

Jitendra Kodilkar, Ishwara Chandra, A.P. Rao

January 13th, 2008

1.Introduction

The ONLINE software controls and monitor the GMRT antenna systems. This software consists of a number of independent communicating programs which runs on Unix workstation. These programs are developed in-house mainly using Fortran and C langauge. The ONLINE is a central supervisory system which communicates, controls and monitor the GMRT's systems and subsystems including servo, antenna receiver, telemetry, and other systems in central electronics building (CEB) like baseband, correlator, pulsar receiver, and data acquisition.

For compilation of online programs, a default compiler (version : SC3.0.1, 1994) in use was accessing a license installed on SunOS server class workstation named *anuradha*. In spite of online being run on newly bought two solaris workstations (*aditya* and *bhaskar*), the access of compiler license on old machine *anuradha* was must while compiling a online software for any modifications required by various streams of users. Since old SunOS machine *anuradha* became obsolete, it was an urgent need to have the ONLINE system compiled with a new independent compiler so that any kind of modification and improvement could be possible with the existing system. Therefore, a new compiler (version : Sun 5.7, 2005) installed on Solaris workstations *aditya* and *bhaskar*. To check correctness test of a new compiler and how it handle real large code bases, an AIPS version 31DEC05 (*Astronomical Image Processing System by NRAO*) compiled successfully on both the solaris machines, and expected behavior verified by running aips.

An user interface of the online system usage a aips shell with customized environment, for this purpose a very early version of aips 15JUL95 has been used. Therefore, aips 15JUL95 compiled first so that required aips libraries can be linked during online compilation. Including AIPS libraries, online uses many static libraries related to astronomical modules, aips utilities, and user input modules. Some of these libraries are locally developed and rest of libraries are from open source code under GNU license. Instead of using already compiled object files, all libraries built with a new compiler so that modification with any module is possible or bug can be traced up to the module level. *In brief, ONLINE is compiled from scratch.* As aips version 15JUL95 was quite old, some header files, structure formats, and few Fortran statements syntactically changed to become compatible with the new Sun Fortran-95 compiler. While compiling online programs, we could overcome many problems ranging from some missing functions or libraries (*like Starlink astronomical lib, GNU readline-5.0 lib*), syntax/routine errors in a few source files (not compatible with the new Fortran or C compiler), segmentation faults while executing some receiver settings, and monitoring related errors etc.

During annual maintenance month of April, 2007, ONLINE test-runs were checked by executing test observations. We released newly compiled online on *aditya* machine in October, 2007 (GTAC cycle 13), and it is running smoothly till date. *The purpose of writing a technical report is twofold, firstly, it was essential to note down even minor or all changes done during online compilation so that occurrence of any particular bug can be traced easily. Secondly, ONLINE was handed over to newgroup (Jitendra Kodilkar and Ishwara Chandra), hence organizing the information about all ONLINE components became necessary.* In this report we have briefly covered the various aspects of ONLINE compilation with a new compiler. In section two, we have given details of the old and new compiler, and in section three the compilation of AIPS, bugs overcame, and procedure to build online environment has been given. Section four describes the libraries required for online, bugs faced while compiling and running of the online, and new modifications and improvements done. Finally, we have discussed about proposed changes and overall summary.

2.New Compiler

Table 1: Information relevant to SUN compiler

#	New	Old
1. Host (a) hostname (b) model	bhaskar, aditya Sun Blade 100 (sun4u group)	anuradha SPARC Server 630 MP
2. Operating system	Solaris 8 (SunOS Release 5.8)	SunOs 5.2 or later
3. Compiler (a) C (b) Fortron†	Sun[tm] Studio 10, C 5.7 2005/01/07 Sun Fortran 95 8.1 2005/01/07	C 3.0.1 SC3.0.1 13 Jul 1994 Fortran 3.0.1 f77: SC3.0.1 13 Jul 1994
4. Path	/opt/SUNWspro/bin	/opt/SUNWspro/SC3.0.1/bin[a]
5. Debugger	Sun Dbx Debugger 7.4 2005/01/07	dbx 3.0.1
6. Supporting Operating System	Solaris OS versions 8,9, and 10 (*Support 64-bit code*)	Solaris(TM) 1.x and 2.x [b]
7. License	Stand alone on both ONLINE hosts.	One License on this machine was accessible for ONLINE hosts.
8. HTML Document	/opt/SUNWspro/docs/index.html	/opt/SUNWspro_old/docs

† The Sun[tm] Studio 10 release no longer includes a separate FORTRAN 77 compiler,f77. Much of the functionality of the f77 compiler is now available with the Fortran 95 compiler. f77 command is a script that calls : f90 -f77 -ftrap=%none

a. Now copy as a backup is available in area /opt/SUNWspro_old/SC3.0.1/

b. Solaris 1.x referes to : SunOS(TM) 4.1.1,2,3,3_U1; Solaris 2.x referes to : SunOS(TM) 5.2 or later.

Table 1 gives detail information of a new compiler from Sun Studio 10, also we have tabulated old compiler information along with it for record keeping. Old compiler currently is not being used. On *aditya* machine compiler was installed in */usr/local* area, it is copied to */opt/SUNWspro* area as per recommendation of system manual.

3. AIPS Installation

AIPS 15JUL95 is being used as a command parser for the ONLINE. As this version has not been updated to the latter released versions of AIPS, it was decided to use 15JUL95 itself to avoid uncertainties with the adaptation of latest AIPS versions. Unlike new aips installation procedure using a single perl script, the installation procedure for old aips versions have to be run in four separate steps (usually referred to as INSTEP). The INSTEP1 shell script figures out machine name, architecture, and then setup all directories and configuration files. The INSTEP2 script compiles all aips subroutines and archive all object files in twelve separate libraries (some of these are linked during online compilation). The latter steps creates executables for all tasks by linking object files created in second step. After completion of these steps, we ran the main AIPS program and verified that it is working by running a few usual aips verbs and tasks.
During the compilation procedure, SUN compiler paths in aips installation scripts are getting set automatically by taking an argument from user, but in a few files it need to be change by editing manually. Therefore, the details of **AIPS 15JUL95** installation paths and names of scripts for compilation are mentioned below :

1. Environment variables	**AIPS_ROOT**=/d1/online1/ontest # *Root directory for aips installation.* **AIPS_DIR**=$AIPS_ROOT/15JUL95 **SYSUNIX**=$AIPS_DIR/SYSTEM/UNIX # *compiler path for Fortran and C for INSTEP1* **SYSLOCAL**=$AIPS_DIR/$ARCH/SYSTEM/$SITE # *architecture-specific directories, shell scripts of compiler settings for Fortran & C* **INC**=$AIPS_DIR/INC # *include area.* **AIPSLIB**=$AIPS_DIR/SOL area # *Static and shared aips library path.* **DA00**=/d1/online1/ontest/DA00/<ONLINE-HOSTS> # *AIPS Data directories.*
2. Compiler option & pre-processor	$SYSUNIX dir: **PP**− Transforms exported code (from aips tarball) in to form suitable for the local compilers (i.e. *.C, *.FOR files to *.c *.f) **FDEFAULT.SH**− The Fortran Compiler command & options for all architectures. **FC**− Shell script for compiling Fortran module under UNIX/AIPS. $SYSLOCAL dir : CCOPTS − C Compiler, ASOPTS.SH − Assembler, LDOPTS.SH − Linker.
3. Data area	**DA00** dir : DADEVS.SH File containing a list of AIPS data directories. This area has been linked to '$HOME/bin/aipsbin.sol' area.
4.System Files	AIPS memory file require to built AIPS environment $HOME/bin/aipsbin.sol/NEW → $AIPSLIB/MEMORY/MED000000;
5. ONLINE adverb	$INC/ONLN2.H contain GMRT adverbs and NON POPS variables.

AIPS Environment building for ONLINE

To create a POPS (*People Oriented Parsing System*) and ONLINE environment system files which are required to recognize various online adverbs (input parameter variables) in AIPS shell, or in case these files are lost or corrupted, there are three utility programs to regenerate required files. The procedure to be followed for this purpose is given below [1] :

[1] More information and details can be find from AIPS install Wizard page on NRAO web-site.

```
> setenv DA00 /home/ontest/bin/aipsbin.sol
> update DADEVS.LIST or $HOME/.dadevs file
> cd $BIN (/home/ontest/bin); CDVER.CSH
> RUN FILAIP    # Creates system files
   Password: # Default is AMANAGER
    # Disks # cat entries/disk ( < 0 =>private catalogs; 15 −100)
    0 0
    # interactive AIPS, # batch queues
    0 0
    # TV # TK devices
    0 0
> RUN POPSGN   # This task initializes POPS memory files 1 through 12
    Enter Idebug, Mname, Version (1 I, 2 A's) (NO COMMAS)
    0 POPSDAT TST
    exit
> RUN POSGN    # $DA00/HELP/ONLN2.HLP POPSG definintion files for adverbs
                # used by the ONLINE.
    0 ONLN2 TST
    exit
```

Above procedure should create and initialize twelve system files in $DA00 area with names like ACD000000;, TPD001001;, PWD000000; etc. With the help of utility **SETPAR**, above parameters related to graphics devices, disks , tapes, aipstv, and printer etc can be customized.

Online system uses the POPS interpretive language for user input. For this purpose, special **adverbs** and **verbs** (program running within a aips shell) for GMRT online system have been added in the POPS environment. Values assigned to adverbs like *destination* and *command* in a particular procedure can construct a command to the specific system of antenna. A number of such online commands to control and monitor various antenna systems have been defined in text files called *NMODPROC.014* and *NMODPROC.01E* in a $RUNFIL area. With the help of RUN file facility of aips, these text files can be transfered to POPS environment so that user can type online commands in a aips shell. To create an aips like online environment, instructions given below have to be followed :

```
> cd $DA00
> move total six system files named with SGD01E*, SGD014*, MSD014000;, and MSD01E000;
to some temporary area.
> ./master.5
    GMRT 1: Enter user Id number
    ? # give 40 for previledged user and 50 for subarray user
    > RUN NMODPROC  # execute the file named NMODPROC.014 for User ID 40,
                     # or NMODPROC.01E for User ID 50
    > exit
```

These commands will regenerate required six systems files in $DA00 area so that adverbs and verbs made for online system can be recognized in POPS environment. After successfull creation of these files, one can open online user input shell by executing ./master.5 with user ID 40 or 50, and then procedures defined for online can be checked by giving command *list <procedure name>*.

3.1 Compilation errors fixed during aips installation

During aips compilation, installation procedure produces log for each instep in $SYSUNIX area. There were compatibility errors related to header file structure and syntax related errors arose from fortran f77 and f90 compiler compatibility. Also, as mentioned earlier, compiler and loader oriented paths in scripts FDEFAULT.SH, CCOPTS.SH, and LTOPS.SH changed manually to /opt/SUNWspro/bin. Following changes are done to accomplish the aips installation :
(i) Tape device selection enum 'MT_ISXY' not found in 'mtio.h' file, checked other header files in /usr/include/<sys> area. Therefore MT_ISXY option related stuff commented respectively in three files */d1/online1/ontest/15JUL95/APL/DEV/UNIX/BELL/SOL/ZTAP2.C, ZTPWA2.C, ZMOUN2.C*
(ii) INSTEP3.LOG : In /d1/online1/ontest/15JUL95/SOL/PREP/FITLD.f file − *NUMIF* integer was saved 'only once' as a structure, line modified to make it as a integer.
(iv) INSTEP4.LOG : Many Format statements were exceeding beyond new line, space and characters number counted and adjusted properly for FORMAT argument e.g. FORMAT(53H,"..") e.g. */d1/online1/ontest/15JUL95/Q/PGM/OOP/BLING.FOR*

4. ONLINE Compilation

As this section addresses issues related to online compilation, the details about source codes' logic or algorithim used is not discussed here[2]. Instead, we must focus on details of software components like compiler being used, required libraries, environment files and programs needed on solaris machine to run ONLINE. A new 'ontest' account created for online compilation has to source a file *$HOME/bin/online.set.5* which sets all requisite online environment variables like operating system version, architecture and defines path for run-files, aips-root, default work area for online, log-file area, system files area etc.
Online version : To compile online source code, a latest version of online code extracted from SCCS (Source Code Control System). It is confirmed that the extracted source code files by SCCS are not significantly differing from source code files present in default working area: *∼onsoft/01dec2002/work3*. Hence, to keep update of latest changes done, source code for compilation copied from *01dec2002/work3* directory to new area *∼ontest/05aug2006/work* of 'ontest' account.
Makefiles : Most of astronomical and aips related modules are written in Fortran language, and unix system related modules are written in C language. All header/include files, C, Fortran module files to build online libraries are in default *work* area (/home/ontest/05aug2006/work) of online hosts. This directory also contains number of makefiles to build online libraries and programs. There are two main makefiles called *makefile.master* and *makefile.general.* Both makefiles usage Fortran Pre-processor (PP) and FC shell-script to build fortran libraries, and f90 compiler to built targets (with $-u$ option to report undeclear variable, $-g$ for debugging, and −O2 for basic block-level optimizations). Depending upon value set for *$OSVER*[3], compiler options, OS specific libraries, and suffix to add at the end of taget names are automatically set in makefiles.
Online programs : The online system configuration is defined in *onconfig.def* file present in *work* area. This file gives a list of processes needed to run online with all other details like process name, it's port ID, online ID and executing permissions. Executables for all these

[2] Please refer document on ABC software implementation by madhura Anturkar for further details.
[3] environment variable for SunOS release level, at present it is 5

processes are build by makefile.master and makefile.general. A brief information about targets build by these makefiles is given as follows :
1. master.5, user[0−5].5 → master.5 : This is a main startup program for ONLINE which initializes shared memory for online data structure, spawns requisite processes given in *onconfig.def*, and establishes socket communication to unixhub.5. After all initial setup, master.5 and its links user[0−5].5 programs invoke POPS command parser which takes input from user terminal and convert into the form that is recognized by final processor for execution.
2. unixhub.5 → commhand.5 : As name suggests, this process works as a central hub which establishes communication with master.5, pktdecode.5, serialsock.5, and subac0.5. Hence all communications of online system route via unixhub process. It receives packet from scctask and send it to pktdecode. Also, it takes commands from master/subac, parses it and forward to scctask.
3. pktdecode.5 : Process waits for packets from commhand, decodes it and write into shared memory. Thus, packet-decode establishes a shared memory area containing a strucutre with all information of the online system so that it can be accessible for other processes like displaying online status on screen or log it on a disk.
4. submon.5 : This process mainly keeps track of overall subarray status of antennas, this includes whether antennas in subarray are configured or communicating, tracking status of array, servo computer status, wind speed etc. Submon process establishes communication with unixhub.5 and decodes messages from it. The decoded status is written in shared memory for online display purposes as well as logged in SUMMARY.LOG file.
5. scctask.5 → serialsock.5 : Scctask connects to *router PC* via socket communication which talks to two communication handlers (called *ComH-1* and *ComH-2* used to broadcast commands to two sub-group of antennas), also it establishes contact with unixhub. This process handles routing of data/command packets between communication handler and unixhub.
6. shmserv.5 : This process acts as a shared memory server, a client program can connect to this process over TCP/IP socket for reading and logging online data.
7. subac1.5, subac[0−5].5 → subac1.5 : Subarray controller (subac) process mainly handles subarray control and monitoring tasks which involves tracking a source, initialization of antenna tracking parameters, monitoring the status of antennnas in the subarray, and data acquisition command for subarray etc. Also, subac process can control activities of antenna through a command file. Subac receives all commands and messages from unixhub process.
8. ondisplay.5 : This is a main online display for users which monitors all activities related to antenna and subarray, this includes status of antenna base computer (ABC), servo computer, monitoring points of receiver chain (IF,LO,RF.), telemetry system, wind status, and sentinal system etc.
9. extr.5, main_ex.5 : Using these off-line utility, user can either retrieve or display back the past status of antenna systems to debug or check the technical problems.
note : Online data structure has been defined in *online.h* C header file, using program **htoinc**, online.h can be converted into fortran compatible header file *online.inc*.

4.1 Required libraries for ONLINE compilation :

Online software handles variety of functionality ranging from astronomical calculation, interprocess communication, user interface, and systems' status display etc. A brief information about libraries which fulfills these functionality is given in table 2. All these libraries are stored in *$HOME/lib/solaris* area.

All libraries used in online compilation are of static type. Online compilation also needs aips subroutines libraries present in $(AIPSLIB) area which were generated during AIPS installation step-two. Libraries given in 2 can be constructed in a one run by using *makefile.online* file which consists of number of make commands to consturct these libraries.

Table 2: Required libraries for ONLINE compilation

Lib name	Information	makefile name
liblocalaips.a	aips related modules to built online environment	makefile.liblocalaips
libaips.a	aips related utilities	makefile.libaips
libreadline.a	history and Emacs & vi editing modes for userinput	†
libipc.a	server and client modules on solaris machine.	makefile.libipc
libkum.a	astro utilities.	makefile.libkum
libvlba.a	library contaning the vlba (astro) routines.	makefile.libvlba
libsla.a	routines concerned with astronomical position and time.	‡
libgen.a	library containg general routines for astro. calculation.	makefile.libgen
libgm.a	library containing the gmodem modules.	makefile.libgm

† The *libreadline.a* updated to the Gnu Readline library, version 5.0.
‡ The *slalib V2.4-12* is a Positional Astronomy Library from the starlink software collection.

4.2 Debugging - ONLINE compilation and run time errors

While compiling ONLINE with a new compiler from Sun Studio-10 and during the run-time of ONLINE, there were number of problems ranging from syntactical errors due to old & new compilers compatibility, environment variables or path not set, library not found, wrong subarray status of online display, and occurrence of segmentation faults in some cases. In order to overcome these problems, the changes done or solutions applied are described below under two categories, first one adresses compilation errors and second half gives summary of run time errors.

(I) Errors during ONLINE Compilation :

(1) libreadline.a : (a) Include file <sys/dir.h> not present, include file name replaced with <sys/dirent.h>. (b) Subroutine *readdir* return type is "strucut dirent ∗", so entry structure variable type casted. Also instead of 'd_namlen', 'd_reclen' used according to the declaration given in 'dirent.h' file.

(2) CMDFRM.FOR : Include files ONLE2.INC and DMSG.INC were not getting added in fortran code *CMDFRM.f* generated by Fortran pre-processor (PP). This problem was occurring due to the number of space characters before the declearation of subroutine names. Problem solved after reducing the number of space characters to six.

(4) SUBAC1.FOR : (a) Space characters adjusted for the lines declearing subroutine names.(b) Character type array *program_id* changed from program_id(0) to program_id(0:), and program_id(14:14) for assignment of null character.

(5) libkum.a, libaips.a : For file 'rset.f', internal fortran compiler error − *gstrgsts* routine return 'ending task', replaced this with "write(*,*) 'ending taks';return".

(6) ReadShm.c, WriteShm.c : Variables *ra_target, dec_target* were undefined in SUBARRAY structure, included in online.h and online.inc header files.

(7) It is found that routines from new_sleep.c are unused, hence while archiving the object files only usleep_.c used.

(8) IPCAIPS.f : CHARACTER∗(∗) CSTR is not used anywhere so commented.
(9) libsla.a : All targets mentioned in makefile.master were not able to link *sla_map_* Starlink library functions, so Starlink library version V2.4-12 downloaded and install on solaris platform. All programs linked properly.
(9) Environment variable changes :
(a) In makefiles, library paths related to variable 'APLSUN' changed to 'APLSOL'
(b) Master program was cripping for invalid AIPS_VERSION, so assigned AIPS_VERSION=$CURR in 'online.set.5'.
(c) submon.5 SUMMARY.LOG path was hardcoded, therefore, /export/home/onsoft/data/ area created on aditya machine so that summary log file can be created.
(d)The LOGPATH in online.set.5 in bin area changed to /d1/online1/online_data/log so that problem of online suspending due to the data log area became full can be avoided.
(II) Debugging of run time errors :
(1) Segmentation fault occurrence during the receiver settings of antenna : In MCMCMD subroutine of CMDFRM.f, while setting 32 or 64 digital mask − 'IBTMP' argument was not provided. (May 9th, 2007).
(2) master.5 shell getting suspended due to control keys C-y or C-Z : Users and master terminal was getting suspended and not recovering again. GNU Readline library version 5.0 downloaded and installed in standard area. After compiling online with readline.a, problem solved. (Oct 15th, 2007).
(3) Servo flags in subac/subar window improper :(a) Subac/Subw display always show *VF or eE* flags, this problem was comming for bytes (10) of servo status replaced bit-operation 'and' function by 'iand'. (b) Interchange of flags between antenna ideal (I) and positioning (Z) : to avoid this mismatch, a subroutine *MonAnts* in SUBMON.f file modified for trak_mode condition statements.
(4) ondisplay.5 : Display command *ondisplay.5* given from˜ontest account display properly all the antennas and their names, but same command given from *work* area had problem − only antennas having addresses 0 − 9 appears in display window. This problem was coming due to the presence of *ante.def* file in work area because *ondisplay.5* program makes mistake of reading wrong addresses from this file.
(5) Shared memory size problem : It is found that shared memory size of newly compiled ONLINE was differring from the old one. This problem was occuring because ONCONFIG environment path in "IPC_AIPS.FOR" file was hardcoded to *01dec2002* work area, so this path changed to *05aug2006* work area.

4.3 New modification/developmental changes

(i) Modification in Sub-array display window (March 10th,2007) Ref- Dr. Ishwara Chandra
During observing/test session, sometimes fringes were not coming because of mis-match between tracking source and the source for which delay tracking and fringe corrections are getting applied in the correlator system. To figure out this problem, sub-array status window (Subacwin,Sacwin) of ondisplay.5 program modified so that 'tracking source' and 'the source at phase center' will be displayed simultaneously in sub-array status window.
(ii) Temperature monitor debugging and correction (May 30th,2007) Ref- Shri. R. Balasubramaniam The temperature monitoring window of ondisplay.5 was showing 54.14 degree C temperature for few antennas. When checked carefully, it found that spurious temperature value was coming because channel zero of MCM-0 was getting null/blank data (due

to the 'MCM-0 mask' change or reset automatically). Temperature monitoring subroutine modified so that antenna will show 'IDLE' or 'MASK?' flag if the MCM 0 is idle or mask changed/reseted.

(iii) Grid pointing procedure improvement and modifications (May 17th, 2007) Ref- Dr. N. Kantharia, Prof. V.K. Kulkarni

To update the corrected pointing offsets (calculated by 'grid' pointing programs) quickly, it necessitates that disable/enable 'command monitor' should work through any user. Accordingly, the source files modified and re-compiled on 'aditya' machine with a new compiler. Also, all Grid-pointing programs and procedure organized and improved so that it can be run easily without any problems. While testing the Grid-pointing run, there was a problem in running pointing procedure (GOPNTG) and command file (which calculates and load the offsets) alternately. On Jun 22nd, 2007, after running a few test run, this problem was rectified by modifying a run-file.

The grid-pointing procedure improved for (a) 'disabling/enabling' of command monitor is possible in any 'user' (b) Change of source name as per azimuth/elevation scan points. (c) Grid pointing procedure after a command file run executes without any problem.

(iv) Common disk usage for two ONLINE hosts, Ref- Dr. Ishwara Chandra

At present, the locally mounted disks on separate ONLINE hosts are available for storing files required to conduct observing sessions like source catelogs, command files etc. Also, to log various systems' data routinely (antenna shell temperature, fps counts, wind, servo data, etc), there are four to six auto cross-mounted disks from other machines across the network. Hence, while switching over between two ONLINE hosts, it creates discontinuity for system log files and confusion in finding out latest observing source catalogs and command files for users. To avoid such complexity, a common disk of 180 GB size for both online-hosts mounted on 'oper2' machine. All routine data logging programs related to fps, telemetry, temperature and wind are modified to log the data in a new area. Also, all gtac observations related files, antenna flag log-files moved to new area with preseving date of creation or modification of these files.

5. Further Work

Existing ONLINE system has been running quite stable and bug free over the past few years. Modifications/improvements are being done whenever need arise to accomodate any new system or changes in antenna systems. In this section, we are discussing few needed tasks which are related to user requirements, re-organising of online dependent programs, preparation of standby systems, and interfacing of new developing system with the ONLINE.

1. Re-organisation of ONLINE dependent software : Since ONLINE is a central supervisory system, there are many software utilites and codes have been developed which are ONLINE dependent. This involves monitoring and automatic logging of systems data routinely, interface with other telescope systems (like correlator, baseband, and pulsar console), logging of antenna faults during observing session, and program required to set particular receiver settings etc[4]. All these programs and scripts are written by number of authors need to be organise in a one common account so that its maintenance and backup can be done systematically.

2. Interface to GSBE backend : The new *GMRT Software Back-End* has been recently

[4] See Apendix-I for ONLINE dependent programs

released as a first version for the test observations. The ONLINE interface to this new system is need to be develop.
3. Standby Router machine : The ONLINE router runs on a stand-alone PC called Jupiter (Cerebra machine), a stand-by system for this need to be prepare.
4. A new ONLINE system on Linux platform : Under the upgradation process of GMRT, antenna base computers (ABC) based on 80186 microprocessor are planned to be replaced by Linux operating system PCs. The Linux PCs will have improved version of ABC software which can meet the requirements of the planned modifications or developments in antenna systems. Hence, it is mandatory to develop a new online system based on linux platform to accomodate overall changes. Toward this, a prototype of ONLINE and ABC software has been developed by Laurent Pommier and A.P. Rao. This prototype can be evolved further to develop a new ONLINE system on Linux platform.

References

NRAO Aips page [http://www.aips.nrao.edu/install.shtml#menu_1]
Star Link Library V2.4-12 [http://star-www.rl.ac.uk/]
GNU readline library version 5.0 [http://tiswww.case.edu/php/chet/readline/rltop.html]

Appendix I : Online Dependent Programs

There are many online dependent programs written by various authors to monitor and log the data from antenna systems. Also, a few important server programs passes ONLINE commands to other systems like correlator, pulsar back-end, and baseband. In order to keep integrity and maintenance easiness, all these programs have been copied in a new account 'odsoft' on both the ONLINE hosts.

1. **dasssrv-dvl** - To configure and acquire astronomical data from Digital Back-End. [*Prof. Jayram Chengalur*]
2. **psrsrv** - Server for Pulsar Backends (IA,PA).[*Dr. B.C. Joshi*]
3. **bbserv** - For baseband settings.[*Ref. AjitKumar*]
4. **gsbe** - Modified version of dassrv for new Software Back-End. [*Ref. Jitendra*]
5. **gridpointing** - Grid-pointing related software (azel,caloff) [*Drs. NGK/S.Roy*]
6. **logflags, readflags**- Online error flags during GTAC observation.[*Ref. Deepak*]
7. **table** - Antenna systems' data logging program.[*Jitendra (Author R.K. Sing)*]
8. **scan1** - Warning alarms during observations.[*Santaji*]
9. **gotosrc** - Automatic positioning and tracking of subarray antennas.[*Santaji*]
10. **servo_events**-Recording and monitoring of servo errors.[*Prof. A.P.Rao*]
11. **servo_read** - Recording of servo packets.[*Prof. A.P. Rao*]
12. **fpscount** - Feed positioning system monitoring related programs.[*manisha*]
13. **los** - create_lo, create_locmd.sh :For band-response test purposes.[*Jitendra*]
14. **newauthif** - pulsarset, newauthif - IF power settings program.[*nanaware*]
15. **bb** - bbgain_addpols, related to baseband gain equalization.[*Ksanjay/sachin*]
16. **lostat** - LOS monitoring program.[*Jitendra*]
17. **telmon** - Routine telemtry monitoring program [*R.Balasubramanian*]
18. **wind** - Routine wind monitoring program.
19. **temperature_alarm**- 'atemp' program.
20. **temperature_log**- 'all_temp' program for rotine monitoring purpose.