aLB

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION
SPONSORED PROJECT INITIATION

Date: November 15, 1978

Project Title: Short Term Analysis Services -SOLO

Project No: G-36-629

Project Director: Dr. Philip Crews

Sponsor: Battelle Columbus Laboratories; Durham Operations; Durham, NC 27709

Agreement Period: From 9/15/78 Until 4/15/79

Type Agreement: Delivery Order No. 1008 (Time & Materials Subcontract under Prime DAAG29-76-D-0100) thru GTRI

Amount: $4,313 Ceiling Price

Reports Required: Summary of Results; Final Report

Sponsor Contact Person (s):

Technical Matters

Contracting Officer's Technical Representative (COTR)
Capt. Stephen R. Ratzel
AIRMICS
313 Calculator Bldg.
Georgia Institute of Technology
Atlanta, GA 30308

Phone: (404) 894-3137

Contractual Matters
(thru OCA)

Mr. Jack Franklin, Contracting Officer
Battelle Columbus Laboratories
Durham Operations
3333 Chapel Hill Blvd.
P. O. Box 8796
Durham, NC 27707

Phone: (919) 489-2366

Defense Priority Rating: None

Assigned to: Information & Computer Sciences (School/Laboratory)

COPIES TO:

- Project Director
- Division Chief (EES)
- School/Laboratory Director
- Dean/Director—EES
- Accounting Office
- Procurement Office
- Security Coordinator (OCA)
- ✓Reports Coordinator (OCA)
- Library, Technical Reports Section
- Office of Computing Services
- Director, Physical Plant
- EES Information Office
- Project File (OCA)
- Project Code (GTRI)
- Other ____________

CA-3 (3/76)

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION

Date: 6/12/79

Project Title: Short Term Analysis Services -SOLO

Project No: G-36-629

Project Director: Dr. Philip Crews

Sponsor: Battelle Columbus Laboratories; Durham Operations; Durham, NC 27709

Effective Termination Date: Period of Performance: 4/15/79

~~Cessation of Accounting Charges~~: Reporting Period: 5/21/79

Grant/Contract Closeout Actions Remaining:

x Final Invoice and Closing Documents
_ Final Fiscal Report
_ Final Report of Inventions
_ Govt. Property Inventory & Related Certificate
_ Classified Material Certificate
_ Other ______________

TERMINATED

Assigned to: Information & Computer Science (School/Laboratory)

COPIES TO:

Project Director
Division Chief (EES)
School/Laboratory Director
Dean/Director—EES
Accounting Office
Procurement Office
Security Coordinator (OCA)
Reports Coordinator (OCA)
Library, Technical Reports Section
Office of Computing Services
Director, Physical Plant
EES Information Office
Project File (OCA)
Project Code (GTRI)
Other ______________

CA—4 (3/76)

G-36-629
TR

REPORT FOR:

BATTELLE COLUMBUS LABORATORIES DELIVERY ORDER NO. 1008
TIME AND MATERIALS SUBCONTRACT UNDER PRIME DAAG29-76-D-0100
GIT RESEARCH PROJECT NO. G-36-629

Dr. Phillip Crews, Project Director
School of Information and Computer Science
Georgia Institute of Technology
Atlanta, Georgia 30332

The views, opinions, and/or findings contained in this report are those of the author and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

REPORT FOR:

BATTELLE COLUMBUS LABORATORIES DELIVERY ORDER NO. 1008
TIME AND MATERIALS SUBCONTRACT UNDER PRIME DAAG29-76-D-0100
GIT RESEARCH PROJECT NO. G-36-629

Dr. Phillip Crews, Project Director
School of Information and Computer Science
Georgia Institute of Technology
Atlanta, Georgia 30332

INTRODUCTION

The purpose of this effort was to gain access to the Brinch Hansen Concurrent PASCAL system, and to gain experience with the concept of transporting virtual machine architecture onto physical hardware configurations. Since AIRMICS is currently involved in research activity with Kansas State University (KSU) in the area of Network Operating Systems (NOS), the availability of the Concurrent PASCAL system in the AIRMICS laboratory would facilitate participation and evaluation of the NOS research effort. The NOS is written largely in Concurrent and Sequential PASCAL, and the availability of these compilers would allow the installation of the KSU product at AIRMICS.

The Concurrent and Sequential PASCAL compilers, written by Brinch Hansen, translate source language into machine code executable by an abstract stack machine. The portability of the PASCAL source programs is based on the implementation of the abstract stack machine on various hardware configurations. Part of the objective of this project was to demonstrate that an operating system such as the Digital Equipment Corporation's IAS could also provide a basis for implementing the abstract stack machine. The feasibility of this concept could have significant implications in the design of NOS for distributed data processing environments, and in the portability of higher order languages.

The Concurrent PASCAL programming language, developed by Brinch Hansen, allows concurrent execution of multiple processes through the use of shared data structures known as monitors and classes. Since no individual process can manage these data structures, the Concurrent PASCAL system utilizes an administrative resource management facility known as SOLO to perform the administrative functions required to support concurrent processing. The SOLO operating system provides the I/O interface necessary between the executing processes and the kernel, and supports utilities such as process managers, text editors, command line interpreters, and the file management activities to make the system usable.

Brinch Hansen's system, SOLO, has been implemented as a stand-alone operating system on PDP-11 series equipment before; however, this effort proposes to operate SOLO in a variety of configurations, but principally as a subordinate task within another host operating system. The benefits accrued from the installation of the SOLO system at the AIRMICS laboratory are threefold. First, further insight into the problems associated with moving a virtual machine and particularly the problems of interfacing the virtual machine to another operating system will be presented. Secondly, a tool will be provided to investigate concurrent programming utilizing the Concurrent and Sequential PASCAL compilers in SOLO. Lastly, the system will substantially increase the ability of AIRMICS to evaluate and participate in the ongoing research activities at KSU in the area of distributed data processing. These activities include the recently completed grant funded by ARO on functionally distributed computer systems, and a follow on effort involving the provision of network and portable operating systems based on SOLO for operation of the distributed system.

Specifically, the stated objectives of the contract are:

a) Install the SOLO operating system as a task within the IAS operating system on the PDP 11/70. Evaluate the resultant effect on the performance of the SOLO system and identify inherent limitations imposed on a SOLO user operating in this mode.

b) Define the mechanisms necessary (i.e., class and/or monitor data types) within SOLO to interface with the network at KSU. This interface will utilize the message system est- tablished under the previous grant at KSU.

APPROACH TAKEN

The strategy for the installation of the SOLO operating system, along with the two PASCAL compilers, encountered two major difficulties. First, no one was readily available at Georgia Tech who had any experience with the use of SOLO or with the abstract stack machine utilized in the system. Secondly, the Concurrent PASCAL system as distributed was designed to run on the hardware of the PDP 11/45. The transiton to the environment presented by IAS would require an intimate understanding of the abstract machine functions such as process initiation, memory allocation, context switching, process management, etc. to insure faithful evaluation by an IAS program. Consequently a three step approach was developed to satisfy the objectives of the effort. Specifically its three steps included: the installation of the SOLO operating system in a stand-alone environment; the installation of SOLO as a task under the IAS operating system; and finally the utilization of SOLO to implement a Concurrent PASCAL solution to the KSU interface. The installation of SOLO in a stand-alone version first was deemed necessary to provide experience with the operation of the system, to identify idiosyncracies in the system, and to develop an appreciation for the difficult aspects in moving the system to IAS.

The installation of SOLO in a stand-alone environment is characterized by two phases. First is the acquisition and analysis of the SOLO operating system distribution material to identify any code that requires modification. Because SOLO is purported to be a portable system, any modification should lie within the kernel interpreter, which contains device-specific algorithms. For the PDP 11/70 installation at AIRMICS at Georgia Tech, discrepencies existed for the disk and tape devices. Specifically, the configuration supported by the distribution system includes a TM11 tape drive and RK05 disk drive, whereas, the AIRMICS system supports TM16 and RP04 devices, respectively. With the appropriate device handler modifications in place in the kernel interpreter, the stand-alone system should be activated through a normal IPL (initial program load) sequence.

Secondly, in anticipation of installing SOLO as a user task under the IAS operating system, a number of modifications to the abstract stack machine are necessary. These entail the transititon from a stand-alone to a multi-user environment. For example, the kernel will not be able to support direct device control, since all I/O devices now become shared resources. Consequently, SOLO must be modified to access virtual devices rather than physical devices. This can be accomplished by substituting system calls, such as QIO's, READ$, and WRITE$, for the device handlers in the kernel.

IAS provides a number of system directives which permit a user program to request file management services directly from the user task. Peripheral devices can be accessed directly through the device handlers of IAS by issuing QIO$ directives which queue an I/O request to a named device. Data transfers are made by the executive through data buffers provided by the user program. The user is provided the capability to read or write logical blocks of data and to receive status information concerning the result of the I/O operation. IAS also provides the capability to read and write virtual data elements such as records through the READ$ and WRITE$ directives. These directives provide access to the file management system of IAS and require additional overhead on the part of the user to open and close the affected files as well as maintain the file descriptor block used by IAS to access system files.

The second step in satisfying the statement of work entails the installation of SOLO as a task under IAS. This requires the creation of virtual devices including a disk, tape drive, console and line printer to be managed by the abstract machine. These virtual devices can be constructed utilizing the system directives. Both the console and tape unit can be simulated through QIO$ directives reading and writing blocks of data at a time.

The 4000 block disk needed by SOLO can either be emulated with QIO$ on the second system disk or represented as a random access file under the file control system. In the first approach, the second IAS system disk would be mounted as a non-sharable resource dedicated to the SOLO user. The disk would be accessed directly by the kernel to transfer blocks of data through the QIO$ directives. The second approach would more efficiently utilize the IAS resources by storing the data of the SOLO disk in a file managed by IAS. Upon initialization of the abstract machine, the file control block for the disk file would be initialized and the file opened as a random access file. The abstract machine would issue READs and WRITE$ directives to transfer virtual blocks of data from the file to the SOLO system.

While the modification of the kernel to exclude priveleged I/O operations with the substitution of IAS directives solves the shared resource problem, another difficulty exists in providing enough task space size to support a SOLO and/or Concurrent PASCAL user. There are two approaches to solving this problem. The first approach is to sufficiently reduce the size of the SOLO operating system, so that more user space is available. This is accomplished in part through substituting calls to the IAS operating system to handle similar functions. The second approach involves establishing the operating system functions in one task partition and the user address space in another partition. The modification required by this approach, in addition to replacing priveleged operations, is the implementation of a mechanism under IAS to support interpartition communication.

The installation of the Concurrent PASCAL System at Georgia Tech creates a conducive environment for the development of an interface to the KSU network. Since the SOLO operating system and Concurrent PASCAL are integral parts of NOS, the expanded IAS provides the only mechanism for any of the Georgia Tech facilities to join the network. Specifically, this phase involves the creation of a node on the PDP 11/70 through the development of a Concurrent PASCAL system to handle NOS communication protocols. This operative node, residing on a different host machine than those at KSU, will provide an interesting research tool in the study of distributed data processing in a heterogeneous network.

The success of this three step approach was predicated on two assumptions. The first assumption involved expecting that only a moderate level of effort would be required to modify the existing abstract machine to operate stand-alone on the AIRMICS PDP 11/70 configuration. Since the system would only run stand-alone temporarily, the disk and tape interfaces which had to be changed should only require minimal code for a device handler. Sophisticated head operations, pre-seeks, error correction, etc., would not be required. The second assumption involved anticipating the message system protocol described in KSU documents would be adequate as the basis of a PASCAL program to achieve the second overall objective.

RESULTS

At the end of the contract period, a bootable, stand-alone SOLO operating system was constructed and the designs for the rest of the project were extensively investigated. While this did not satisfy the overall objectives of the statement of work, it provided a strong foundation for the continuation of the effort. In general, the unforseen level of effort to port a software system to an incompatable host limited the amount of the statement of work that could be fulfilled, although the exercise to date did provide insight that will further direct the rest of the design effort. The major problems involved with the software transport included: incomplete documentation, an incompatible execution environment, and, limited system access. Of the three difficulties incomplete and inaccurate documentation hampered the installtion of the initial SOLO most severely. Undocumented items had to be subsequently dealt with included SOLO bootstrap loader, the physical organization of the logical blocks of the system on the distribution tape, and general vagueness of the machine dependent sections of the kernel.

Of course the difficulties encountered with inaccurate documentation speak for themselves. Many of the difficulties associated with the documentation forced the investigation team to write utility programs to map the distribution tapes into the PDP 11/70 environment. Though seemingly trivial, the development of programs, such as tape copy to disk, disk loader/ tape loader, and disk overlay were extremely time-consuming because they had to be tested without the operating system, due again to the rigid operational specification of SOLO. For example, for the tape to disk copy procedure, the target area of the disk was fixed and conflicted with IAS access, a function of security of the PDP 11/70 operating system. Consequently, program debug and check out had to be accomplished through the console switches. Furthermore, the development of those programs was hampered by the lack of an absolute loader, and while they could be entered under IAS, additional software was required to extract the task header from the resulting load module, so that the program could be run in the native machine mode. This difficulty in construction of pure (machine) code made it particularly difficult to develop and test the device handlers, for the RP04 and TM16, that had to be inserted into the kernel code. Finally the requirement to even operate the machine in the native mode conflicted with the guaranteed operation of the IAS and UNIX operating system, and thus system access was somewhat limited.

While the statement of work was not completely satisfied, an initial design was developed to satisfy the other requirements of the project. However, even these designs, which were conceptually very sound, were found to be prone to difficulty, due to incompatibility of IAS and the SOLO system requirements. For example, one approach described above was to reduce the size of the kernel/interpreter so that it could be installed as a task under IAS, and consequently provide a larger user address space. Unfortunately, SOLO appears to use separate I/D space (instruction/data) whereas IAS uses only I space. As a result, when SOLO is initiated IAS will abnormally terminate. In the alternate approach the kernel and user space are separated under IAS, and a communication path must be established between the SOLO operating system and the user. Again, IAS version 2 (which is currently operating at the AIRMICS installation) provides only an intertask communication mechanism which is extremely cumbersome and difficult to use.

CONCLUSION

The purpose of this analysis was to provide access to the Concurrent Pascal System in the AIRMICS Research and Development Laboratory in order to facilitate the interaction of AIRMICS with the distributed processing research at Kansas State University. An implementation strategy, consisting of three steps, was chosen to satisfy the specific objectives of the contract. The execution of the three step approach appeared to be feasible, within the time constraint of the contract, under the following initial assumptions:

a) Only a moderate level of effort would be required to modify the existing abstract machine to operate stand-alone on the AIRMICS PDP 11/70 configuration.

b) The message system protocol from the KSU documents would be adequate as a basis for a PASCAL interface into the NOS.

Although significant progress was made toward fulfillment of the stated objectives of the contract, the limited time available did not provide any flexibility to deal with the unforseen difficulties that arose with these initial assumptions. In light of the problems that hampered the project, the work that was completed represents a major step toward the installation of the Concurrent PASCAL System. As a result of the development of this initial basis of operating software and increased knowledge of the system, the researcher feels confident that the project can be successfully completed with additional funding.

G-36-629

G-36-627

BATTELLE COLUMBUS LABORATORIES – DURHAM OFFICE – STAS CLAIMS VOUCHER

Claims incurred in connection with scientific services rendered under Delivery Order No. 1008

me of Payee Georgia Tech Research Institute ~~Social Security No.~~ FEI 58-0603146

id Check to Georgia Tech Research Institute Start Date ______ Return Date ______

dress Georgia Institute of Technology Destination ______

Atlanta, Georgia 30332

CLAIM FOR SCIENTIFIC SERVICES (COTR VERIFICATION REQUIRED)

From 9/15/78 (Beginning Date) to 2/28/79 (Ending Date) – 14 days at $ 203.00 per day $ 2,842.00

(A breakdown by dates should be given if days were not worked consecutively.)

:RIFICATION FOR AUDIT PURPOSES: The undersigned Contracting Officer's Technical Representative verifies that services have be idered as specified above.

)TR Signature & Title ______ Date ______

Capg. Stephen R. Ratzel COTR

TRAVEL & OTHER EXPENSES PAID BY TRAVELER (COTR VERIFICATION NOT REQUIRED)	TOTAL
)TEL (Room & Tax Only – Attach Receipts)	
EALS & TIPS	
ISC. (Tips, Baggage, Laundry - specify)	
)MMERCIAL CONVEYANCE (Plane, Train, Bus–Attach all ticket stubs)	
RAVEL IN PERSONAL CAR (miles at 17 cents =)	
ENTED CAR EXPENSE (Deduct personal use portion–attach receipts)	
AB & CAR FARE (Breakdown if over $20–Give details)	
ELEPHONE & TELEGRAPH (Breakdown if over $20–Give details)	
THER (Itemize & furnish receipts for items of $20 or more)	
TOTAL TRAVEL & OTHER EXPENSES	
TOTAL AMOUNT CLAIMED (Sections I and II)	2,842.0(
ADVANCE	
CASH SETTLEMENT	2,842.0

declare the contents of this form are correct and true to the best of my knowledge and belief.

ate MAR 23 1979 Signature of Payee ______

Dwight L. Allen, Contracting Officer

OTE: The reverse of this voucher should be used in submitting to this office an unclassified summary of results of services provided. reserves the right to refuse payment of this voucher unless the required summary report is furnished.

ATTELLE APPROVAL: ______ (Administrative Officer) ______ (Program Manager) ______ (Date)

heck Date ______ Code No. ______