C^H Shell for Integration of Mechatronic Systems

Harry H. Cheng
Department of Mechanical and Aeronautical Engineering
University of California
Davis, CA 95616

Abstract

C^H is a very high level propgramming language, it is a superset of ANSI C. C^H combines C, Fortran, robot languages, and Unix shells into a single programming language. It has been used as a real-time Unix shell in a VMEbus-based distributed computing system for integration of mechatronic systems for agile manufacturing.

1 Introduction

Agile manufacturing is charactered by its ability to allow a rapid response to continuously changing customer requirements and markets. Software is the key to the agility of many flexible manufacturing systems. For many tasks in manufacturing and automation, high-performance sophisticated computationally expensive control is not as important as ease of integration of mechatronic devices that include stepper motors, DC and AC motors, laser and vision sensors, force and torque sensors, tactile sensors, and pneumatic grippers [1]. These mechatronic devices are essential components of robot manipulators, CNC machines, and automatic assembly systems. Computerized integration of mechatronic devices is one of major efforts in development of flexible manufacturing and automation systems. Typically, these devices are integrated with PCbus or VMEbus-based distributed computing systems in C programming language. For different type of applications, user interfaces to machinery are different, even if the underlying mechatronic devices and operation principles are the same. Significant recoding is often neeeded in response to a small change of customer requirements, which implies more cost and longer delivery time. For agile manufacturing, what the industry needs is a Unix-based open architecture high level programming environment that can be used for various industry applications and integration of mechatronic systems.

2 C^H Shell

We have developed the C^H programming language for integration of mechatronic systems and consistent interfaces external to machinery. C^H is a very high level programming language, it is a superset of ANSI C with many high level extensions such as addition of complex, dual, and string data types, polymorphism, nested functions, and computational arrays of variable length [2-7]. C^H is also an interpretive Unix shell with a quick system response [8]. Therefore, integration of mechatronic systems for agile manufacturing can draw on the large existing body of skilled C and Unix users. C^H has been used for Unix and real-time Unix shell programming. C^H shell is similar to C shell in handling of variables. Both C^H and C shells have their local shell variables besides the environmental variables. Table 1 gives a brief syntax comparison between C^H and C shells. Semantics of C^H commands listed in Table 1 are the same as those of the corresponding commands of C shell. However, unlike the so-called C shell, C^H is a superset of ANSI C. The programming capabilities of C^H shell go far beyond those of C shell. C^H is appealing to both seasoned programmers and novice casual users. Our experience indicates that sophisticated users may write task-level functions and toolboxes that can be readily used by less experienced users [8].

C^H has been used for shell programming under a real-time Unix operating system LynxOS in a VMEbus based distributed computing system. Integration of mechatronic devices for agile manufacturing in C^H is currently under investigation. C^H will be configured for user interface of different mechatronic systems in the same manner as Unix kernels are built for interface of different computer hardware platforms.

3 References

1. Cheng, H. H., Real-Time Manipulation of a Hybrid Serial-and-Parallel-Driven Redundant Industrial Manipulator, *ASME Journal of Dynamic Systems, Measurement, and Control*, December, 1994.
2. Cheng, H. H., Scientific Computing in the C^H Programming Language, *Scientific Programming*, Vol. 2, No. 3, Fall, 1993, pp. 49-75.
3. Cheng, H. H., Handling of Complex Numbers in the C^H Programming Language, *Scientific Programming*, Vol. 2, No. 3, Fall, 1993, pp. 76-106.

Table 1: Selected syntax comparisons between C^H and C shells.

C^H shell	C shell
`_prompt = "ch> "`	`set prompt = "ch> "`
`_path = "/usr/bin /bin"`	`set path = (/usr/bin /bin)`
`umask(022)`	`umask 022`
`putenv("PATH=/usr/bin /bin")`	`setenv PATH "/usr/bin /bin"`
`printf("%s\n", getenv("PATH"))`	`echo $PATH`
`echo $(getenv("PATH"))`	`echo $PATH`
`printf("%s\n", _path)`	`echo $path`
`echo $_path`	`echo $path`
`remenv("PATH")`	`unsetenv PATH`
`remvar("_path")`	`unset path`
`int i = 90`	`set i =90`
`i = 91`	`set i =91`
``string _host = `hostname` ``	``set host = `hostname` ``
``_host = `hostname` ``	``set host = `hostname` ``
`alias("rm", "rm -i")`	`alias rm 'rm -r'`
`unalias("rm")`	`unalias rm`
`ls ~ *`	`ls ~ *`
`ls > output`	`ls > output`
`ls | tar -cf tarfile`	`ls | tar -cf tarfile`
`ls $(getenv("PATH"))`	`ls $PATH`
`ls $_path`	`ls $path`
`/usr/bin/ls *`	`/usr/bin/ls *`
`./cmd -option`	`./cmd -option`
`_histsize = 100`	`history = 100`
`history`	`history`
`$l`	`!l`
`$l -agl`	`!l -agl`
`$3`	`!3`
`$-1`	`!-1`
`$$`	`!!`
`grep "str1 str2" -l *.c | vi`	`grep "str1 str2" -l *.c | vi`
`more .chshrc .chlogin .chlogout`	`more .cshrc .login .logout`

4. Cheng, H. H., Programming with Dual Numbers and Its Applications in Mechanisms Design, *Engineering with Computers, An International Journal for Computer-Aided Mechanical and Structural Engineering*, Vol. 10, No. 4, 1994, pp. 212-229.
5. Cheng, H. H., Extending C and FORTRAN for Design Automation, *Proc. of ASME Design Automation Conference*, Minneapolis, MN, Sept. 11-14, 1994, Vol. 1, pp. 93-103.
6. Cheng, H. H., Extending C with Arrays of Variable Length, WG/N348, X3J11/94-033, June 6, 1994; also *Computer Standards and Interfaces* (in press).
7. Cheng, H. H., Adding References and Nested Functions to C for Modular and Parallel Programming, NCEG, X3J11.1/93-044, October 22, 1993.
8. Cheng, H. H., Pedagogically Effective Programming Environment for Teaching Mechanism Design, *Computer Applications in Engineering Education*, Vo. 2, No. 1, 1994, pp. 23-39.

4 Acknowledgements

The author would like to thank Delta Tau Data Systems, Inc. for contributing two eight-axes VME servo control boards and one VME I/O board to this project.