

EARTH OBSERVATION SYMPOSIUM (B1)
Interactive Presentations (IP)

Author: Mr. Zhao Wang
Chinese Academy of Sciences, China

Mr. Chuan You
China

Prof.Dr. KaiLiang Han
Chinese Academy of Sciences, China

Mr. WeiXin Cui
China

Ms. Sha Luo
China

A DYNAMIC SOFTWARE RESTRUCTION METHOD BASED ON AT697

Abstract

AT697 is a widely used CPU in spacecraft electrical facilities, which undertakes most of the data computation by executing the software. To fulfill the need of space application, sometimes the software must be restructured to alter the function of some special apparatus or fix errors without stopping its main function, for the space instruments are expected to maintain working all the time. Some methods were proposed, refreshing the whole or parts of the code in ROM and then resetting the system to reload the new code. These methods were called static software restruction methods with which the main task must be stopped and the system must be restarted again. The restruction method must be dynamic to avoid stopping the main task of the instrument. In this paper, a new restruction method is introduced to alter the code segment in RAM while the main task is still being executed. Only the fixed parts of code are suspended temporarily and resetting the system to load new code from ROM is not necessary. Using binary code to override the code segment is the most reliable method. However, it will be a complicated work for software designer to fix intricate functions. Since the tasks of space application are becoming more and more complex and diversiform, even using assembly language to rewrite the target code will be an arduous work. The proposed method uses C language to rewrite the functions to make this work easier. Software designers fix only target part of the origin code using C language, converting the code into binary code, supplying the fixed code to software restruction function, and then the function override the target part in RAM automatically. All the data in RAM including code segment, data segment and stack segment must be allocated properly. The address of each function, variable will be the target address to override itself. There must be enough room left at the tail of each function, for the new function may occupy more memory. A software restruction strategy to arrange memory allocation will be introduced in this paper, using IDE tools. Key Words: AT697, dynamic software restruction, variable address, space application, C language