

Analysis and Optimization of Program Size, Program Validation and Testing

Analysis and Optimization of Program Size

Data provide an excellent opportunity for minimizing size because the data are most highly dependent on programming style.

In data dominated applications, such as image or speech signal processing applications, summing up the sizes of all the arrays is the most straightforward way to get an upper bound of the memory requirement.

In the data dependency relations in the code are used to find the number of array elements produced or consumed by each assignment, from which a memory trace of upper and lower bounding rectangle as a function of time is found.

Care should be taken while designing buffer size. Data can sometimes be packed, such as by storing several flags in a single word and extracting them by using bit-level operations.

A very low-level technique for minimizing data is to reuse values. Data buffers can often be reused at several different points in the program.

Minimizing the size of the instruction text of a program requires a mix of high-level program transformations and careful instruction selection.

Encapsulating functions in subroutines can reduce program size when done carefully.

Program Validation and Testing

Testing is an organized process to verify the behavior, performance, and reliability of a device or system against designed specifications.

Debugging is the process of removing defects ("bugs") in the design phase to ensure that the synthesized design, when manufactured will behave as expected. Testing is a manufacturing step to ensure that the manufactured device is defect free.

Embedded software development uses specialized compilers and development software that offer means for debugging. Developers build application software on more powerful computers and eventually test the application in the target processing environment.

Testing methods are of two type :

1. **Black-box testing :** This method generates tests without looking at the internal structure of the program.
2. **White box testing :** This method generate tests based on the program structure. This method also called as Clear-box testing.

Black Box Testing

Black box testing is also called functional testing. It is testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions.

With black box testing, the software tester does not have access to the source code itself. The code is considered to be a "big black box" to the tester who can't see inside the box.

Black-box is based on requirements and functionality, not code.

Random tests form one category of black-box test. Random values are generated with a given distribution.

The expected values are computed independently of the system, and then the test inputs are applied. A large number of tests must be applied for the results to be statistically significant, but the tests are easy to generate.

Using black box testing techniques, testers examine the high-level design and the customer requirements specification to plan the test cases to ensure the code does what it is intended to do.

Functional testing involves ensuring that the functionality specified in the requirement specification works. System testing involves putting the new program in many different environments to ensure the program works in typical

customer environments with various versions and types of operating systems and/or applications.

Advantages :

1. Tests the final behavior of the software.
2. Can be written independent of software design.
3. Can be used to test different implementations with minimal changes.

Disadvantages :

1. Doesn't necessarily know the boundary cases.
2. Can be difficult to cover all portions of software implementation.

White Box Testing

Often called "structural" testing.

Knowing the internal workings of a product, test that all internal operations are performed according to specifications and all internal components have been exercised.

It involves tests that concentrate on close examination of procedural detail. Logical paths through the software are tested.

White box testing focuses on the internal structure of the software code. The white box tester knows what the code looks like and writes test cases by executing methods with certain parameters.

Test cases exercise specific sets of conditions and loops.

A white-box testing technique that focuses exclusively on the validity of loop constructs. Four different classes of loops exist : Simple loops, nested loops, concatenated loops and unstructured loops.

Advantages :

1. Usually helps getting good coverage.
2. Good for ensuring boundary cases and special cases get tested.

Disadvantages :

1. Tests based on design might miss bigger picture system problems.
2. Tests need to be changed if implementation/algorithm changes.
3. Hard to test code that isn't there (missing functionality) with white box testing.