

实验三 白盒测试实验三

一、实验目的

- (1) 巩固白盒测试知识，能应用数据流覆盖方法设计测试用例；
- (2) 学习测试用例的书写。

二、实验环境

硬件环境：PC 机一台

软件环境：Java 编程环境：Java SDK + Eclipse

C/C++编程环境：Visual Studio

程序流图绘制：Visio

待测程序：CgiDecode

实验指导书、Eclipse 和待测程序可从 <FTP://223.3.68.135> 或课程主页下载：

<http://cse.seu.edu.cn/PersonalPage/pwang/course/st.html>

三、实验内容

实验背景：

在 Web 服务等应用中，CGI (Common Gateway Interface) 是用户访问服务器端 Web 页面内容的一种传输标准。

有关 CGI 的文档详见：

http://en.wikipedia.org/wiki/Common_Gateway_Interface

<http://tools.ietf.org/html/rfc3875>

<http://baike.baidu.com/view/32614.htm>

在应用程序开发中，常常需要将 CGI 编码的字符串解码为普通的 ASCII 字符串。

本次实验的被测程序 CgiDecode 展示了此功能的 C 语言实现。

实验 1：数据流测试技术实验

运用数据流测试方法，对用 C/C++语言实现的 CgiDecode 程序中的 decode()方法进行测试。

要求：

- (1) 测试要考虑 decode()中 encoded, decoded, *eptr, eptr, *dptr, dptr, ok, c, digit_high, digit_low 变量；
- (2) 给出每个变量对应的 du-path 和 dc-path；
- (3) 根据变量的 dc-path 设计测试用例，完成对 decode()的测试；

实验过程注意事项：

- 1. 变量*eptr 和变量*dptr:
由于这种变量涉及到对指针进行*操作，因此非声明位置出现*eptr 和*dptr 的时候都视为是相应指针 eptr 和 dptr 的使用节点。
- 2. 难点 1：正确分析变量的定义节点和使用节点；
- 3. 难点 2：变量的定义节点不要求变量一定出现。
在指针发生变化的时候，会影响到相应的指针变量的值。因此，语句 22 虽然没有出现*eptr，但却是*eptr 的一个定义节点。
- 4. 提供一个 CGI 解码的程序供理解和测试过程中参考，其中 getHexValue()的作用是取对应字符串的十六进制值。
- 5. DU-PATH/DC-PATH 的数量及其确定：定义节点 A 到使用节点 B 之间可能有多条 DU-PATH/DC-PATH，理论上这些 DU-PATH/DC-PATH 都需要进行测试，如果这些路径可能有无穷多条，请指出来。

decode()函数的语句及其编号如下：

1	/** Translate a string from the CGI encoding to plain ascii text.
2	* ' ' becomes space, %xx becomes byte with hex value xx,
3	* other alphanumeric characters map to themselves.
4	* Returns 0 for success, positive for erroneous input
5	* 1 = bad hexadecimal digit
6	*/
7	int decode(char *encoded, char *decoded)
8	{
9	char *eptr = encoded;
10	char *dptr = decoded;
11	int ok=0;
12	while (*eptr)
13	{
14	char c;
15	c = *eptr;
16	if (c == '+')
17	{ /* Case 1: '+' maps to blank */
18	*dptr = ' ';
19	}
20	else if (c == '%')
21	{ /* Case 2: '%xx' is hex for character xx */
22	int digit_high = getHexValue(++eptr);
23	int digit_low = getHexValue(++eptr);
24	if (digit_high == -1 digit_low== -1) {
25	/* *dptr='?' ; */
26	ok=1; /* Bad return code */
27	} else {
28	*dptr = 16* digit_high + digit_low;
29	}
30	} else { /* Case 3: All other characters map to themselves */
31	*dptr = *eptr;
32	}
33	++dptr;

34	++eptr;
35	}
36	*dptr = '\0'; /* Null terminator for string */
37	return ok;
38	}

实验 2：白盒测试工具的使用（业余时间选做）

白盒测试需要耗费大量的时间，而白盒测试工具能在一定程度上降低测试工作量。

请尝试使用课堂中介绍过的白盒测试工具对实验 1-2 中的程序代码进行测试，体会静态白盒测试工具和动态白盒测试工具的用途。

建议尝试或了解 Parasoft JTest 或 Parasoft C++Test 工具的白盒测试功能。Parasoft C++Test 的某个早期版本参见 FTP 中实验目录下的软件。Parasoft JTest 可从官方下载评估版本。

这些商业化测试工具都很复杂，建议安装和使用过程在虚拟机中进行，以免影响本机的其它软件和操作系统。

建议收集并使用一些开源的白盒测试工具。

四、实验要求

- (1) 编写的测试用例要合理、清晰。
- (2) 撰写实验报告（参照实验报告模板）
- (3) 实验报告命名规则：**学号-实验次序.pdf**

五、实验思考

- (1) 通过测试，是否发现程序中存在的缺陷？
- (2) 谈谈数据流测试和控制流测试的区别和联系。
- (2) 如果用工具来替代手工的白盒测试，你觉得这样的工具应该如何设计？设计的技术中可能的技术难点在哪里？

六、附录