บทที่ 5
การทดสอบและผลการทดสอบ

เนื้อหาในบทนี้เป็นการทดสอบเพื่อประเมินผลโครงงาน ว่าสามารถบรรลุวัตถุประสงค์ของโครงงานที่ได้ตั้งไว้ได้หรือไม่ โดยได้มีการตั้งวัตถุประสงค์ของการทดลองดังนี้

1. เพื่อให้สามารถพัฒนาบริการบนระบบที่ได้สร้างขึ้นได้
2. เพื่อทดสอบการให้บริการของเว็บเซอร์วิสบนระบบฝังตัว
3. เพื่อทดสอบการทำงานของระบบแบบมัลติทาสก์กิ้งและเรียลไทม์

ในการทดสอบโครงงานนี้ จะแบ่งการทดสอบออกเป็นสองส่วนคือ การทดสอบการทำงานของระบบและการทดสอบการใช้งานของระบบ

5.1 การทดสอบการทำงานของระบบ

เป็นการทดสอบเพื่อแสดงให้เห็นว่าระบบนี้สามารถทำงานได้ตามที่ได้ออกแบบไว้หรือไม่ โดยแยกหัวข้อการทดสอบและผลการทดสอบดังตารามด้านล่าง

การทดสอบ	**ผลการทดสอบ**
การทำงานแบบมัลติทาสก์กิ้ง	ทาสก์ในระบบสามารถทำงานร่วมกันแบบมัลติทาสก์ได้
การสื่อสารข้อมูลผ่านระบบเว็บเซอร์วิส	สามารถติดต่อสื่อสารในระบบเว็บเซอร์วิสผ่านเครือข่ายอินเตอร์เน็ตได้อย่างถูกต้อง

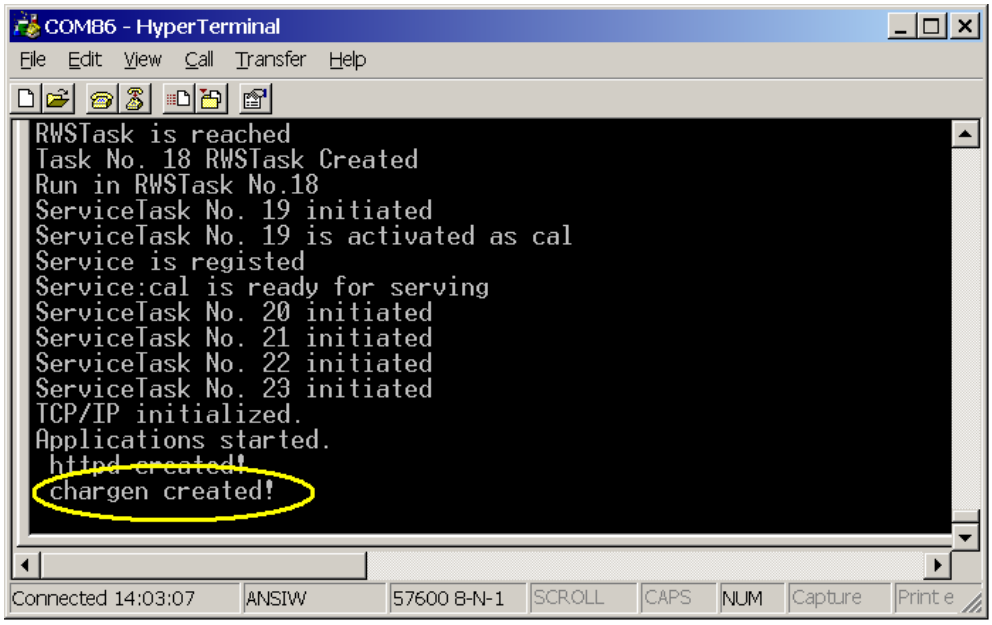
ตาราง 5-1 ผลการทดสอบการทำงานของระบบ

รายละเอียดของการทดสอบการทำงานในแต่ละหัวข้อจะได้อธิบายดังต่อไปนี้

5.1.1 ทดสอบการทำแบบมัลติทาสก์กิ้ง

การทดสอบการทำงานแบบมัลติทาสก์กิ้ง เพื่อทดสอบว่าระบบสามารถรองรับการทำงานของทาสก์บนระบบพร้อมๆกัน และมีความเป็นอิสระต่อกัน แต่เนื่องจากการทำงานแบบมัลติทาสก์กิ้งนั้นเป็นกลไกการทำงานภายใน ไม่สามารถทดสอบการทำงานโดยตรงได้ จำเป็นที่จะต้องแสดงการทำงานผ่านหน้าจอ หรือ ทดสอบการติดต่อผ่านระบบเน็ตเวิร์คที่สามารถรองรับการเชื่อมต่อได้มากกว่า 1 การเชื่อมต่อพร้อมๆกัน

ตัวอย่างที่ใช้ในการทดสอบการทำงานแบบมัลติทาสก์กิ้ง เป็นการสร้างบริการผ่านระบบเน็ตเวิร์ค ด้วยโปรโตคอลชาร์เจน (Chargen) บนพอร์ตการสื่อสารหมายเลข 19 โดยสร้างโปรแกรมที่ทำงานเป็นเซิร์ฟเวอร์ที่รองรับการเชื่อมต่อ 5 การเชื่อมต่อพร้อมกัน โดยหลังจากสั่งให้โปรแกรมทำงาน จะมีลักษณะดังรูปด้านล่าง

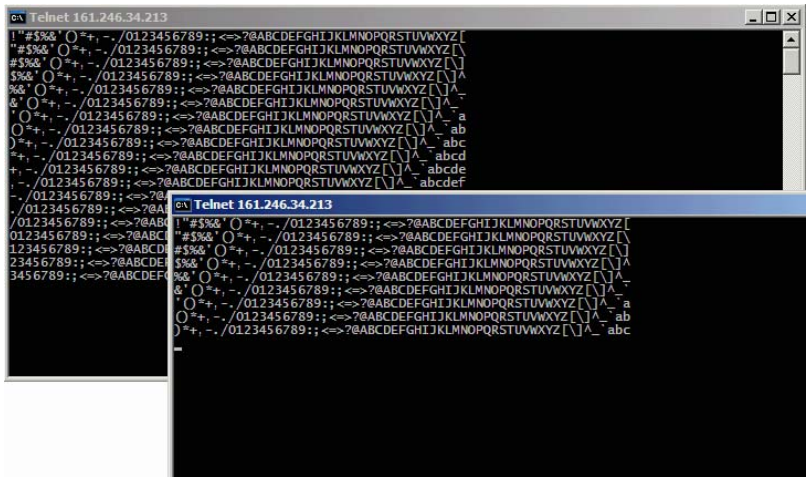


รูปภาพ 5-1 การเขียนโปรแกรมทดสอบการทำงานแบบมัลติทาสก์กิ้ง

หลังจากสั่งการให้โปรแกรมที่ทดสอบทำงาน ก็ทำการขอใช้บริการโดยทำการเชื่อมต่อด้วยโปรแกรมเทลเน็ต (telnet) ที่มีอยู่บนระบบปฏิบัติการวินโดวส์ (Windows) โดยใช้คำสั่งซึ่งมีรูปแบบดังนี้

telnet <ip-address> <port number>

โดยให้ ip-address เป็นหมายเลขไอพีของอุปกรณ์คอม86 และหมายเลขพอร์ตเป็นหมายเลข 19 โดยเปิดโปรแกรมเทลเน็ตจำนวน 2 โปรแกรม แล้วป้อนคำสั่งดังกล่าว แล้วให้ทำงานพร้อมๆกันเพื่อทดสอบผลการทำงานแบบมัลติทาสก์กิ้ง โดยผลการทดลองเป็นไปดังรูปต่อไปนี้

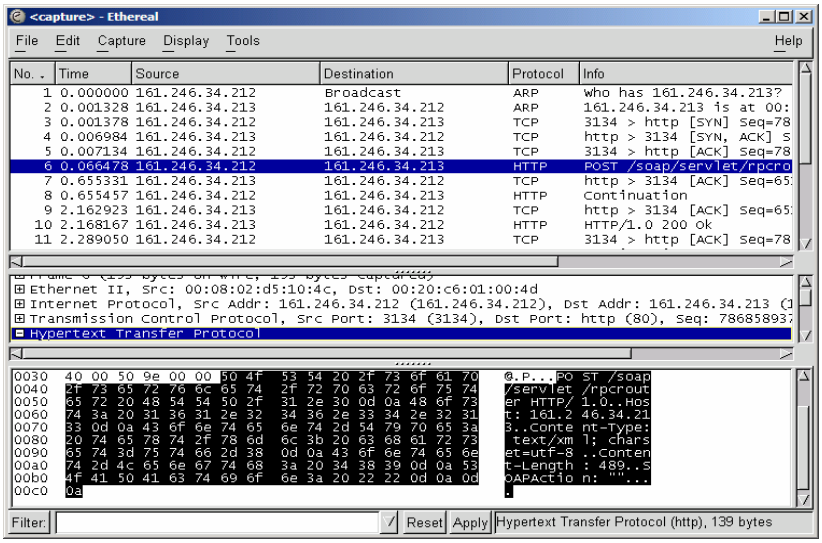


รูปภาพ 5-2 แสดงการติดต่อขอใช้บริการโปรแกรมที่ทำงานแบบมัลติทาสก์กิ้งบนอุปกรณ์คอม86

ผลการทดลอง ทำให้รู้ได้ว่า สามารถสร้างโปรแกรมที่ทำงานแบบมัลติทาสก์กิ้งบนอุปกรณ์คอม 86 ได้โดยแสดงในรูปของการรองรับการเชื่อมต่อขอใช้บริการผ่านระบบเน็ตเวิร์ค 2 การเชื่อมต่อพร้อมกัน

5.1.2 ทดสอบการสื่อสารข้อมูลผ่านระบบเว็บเซอร์วิส

การทดสอบการสื่อสารข้อมูลผ่านระบบเว็บเซอร์วิส ทำได้โดยทำการติดตามและดักจับข้อมูลที่ส่งผ่านเครือข่ายที่ถูกส่งออกมาขณะที่ระบบเว็บเซอร์วิสมีการติดต่อขอใช้บริการ โดยการติดต่อข้อมูลในระบบเว็บเซอร์วิสนั้น อยู่บนโปรโตคอลเอชทีทีพี ที่อยู่บนโปรโตคอลทีซีพี/ไอพี โดยอยู่ใช้หมายเลขพอร์ต 80 โดยผลการดักจับข้อมูลในการติดต่อสื่อสารข้อมูลขณะใช้งาน โดยใช้โปรแกรมอีเธอร์เรียล (Ethereal) มีดังรูปต่อไปนี้



รูปภาพ 5-3 ผลการตรวจสอบการรับส่งข้อมูลในระบบเว็บเซอร์วิสบนคอม86 โดยโปรแกรมอีเธอร์เรียล

5.2 การทดสอบการใช้งานของระบบ

เป็นการทดสอบผลการใช้งาน ว่าสามารถทำงานได้ตามที่ออกแบบไว้หรือไม่ โดยแยกหัวข้อการทดสอบและผลการทดสอบดังตารางด้านล่าง

การทดสอบ	ผลการทดสอบ
การพัฒนาบริการบนระบบเว็บเซอร์วิส	สามารถทำการพัฒนาบริการบนระบบเว็บเซอร์วิสได้ตามแนวทางและข้อแนะนำที่กำหนดไว้
การใช้บริการบนระบบเว็บเซอร์วิส	สามารถใช้บริการผ่านโปรโตคอลโซบในระบบเว็บเซอร์วิสได้

ตาราง 5-2 ผลการทดสอบการใช้งานของระบบ

5.2.1 ทดสอบการพัฒนาบริการบนระบบเว็บเซอร์วิส

การทดสอบการพัฒนาบริการบนระบบเว็บเซอร์วิสที่ได้พัฒนาขึ้น ทำได้โดยทดลองพัฒนาบริการขึ้นเพื่อทดลองใช้งานจริง เพื่อทดสอบแนวทางและรูปแบบที่ได้ออกแบบ ในการพัฒนาบริการในลักษณะของเซอร์วิสโค้ด ในการทดลองนั้น ขอยกตัวอย่างการพัฒนาบริการ Calculator ที่ทำการประมวลผลจำนวนจริง ผ่านระบบเว็บเซอร์วิสบนระบบอุปกรณ์ฝังตัว

ในการเขียนโปรแกรมในส่วนของบริการนั้นมีขั้นตอนดังนี้

- เขียนโปรแกรมในภาษาซีพลัสพลัส แล้วพัฒนาเซอร์วิสโค้ดโดยการทำการ inherit โค้ดแม่แบบหรือซุปเปอร์คลาส (Super Class) ที่มีอยู่แล้วในระบบ แล้วทำการพัฒนาเพิ่มในส่วนที่จำเป็นต่อการทำงานของเซอร์วิสโค้ดที่จะพัฒนานั้น ซึ่งได้แก่ ฟังก์ชั่น init และฟังก์ชั่น run

```
void Calculator::init(RPCParam *p_param,RPCParam *p_result){
        param=p_param;
        result=p_result;

        SERVICE_METHOD=param->valueOf((char *)ServiceConsts::SERVICE_METHOD);
        printf("SERVICE_METHOD = %s\n",SERVICE_METHOD);
        ARG1=atof(param->valueOf(ARGUMENT_NAME_1));
        ARG2=atof(param->valueOf(ARGUMENT_NAME_2));
        }
```

รูปภาพ 5-4 การเขียนเซอร์วิสโค้ดในส่วนของฟังก์ชั่น init

```
void Calculator::run(){
	if (strcmp(SERVICE_METHOD,PLUS)==0)		{
		ARG1+=ARG2;
	}
	else if (strcmp(SERVICE_METHOD,MINUS)==0)	{
		ARG1-=ARG2;
	}
	else if (strcmp(SERVICE_METHOD,TIMES)==0)	{
		ARG1*=ARG2;
	}
	else if (strcmp(SERVICE_METHOD,DIVIDE)==0)	{
		ARG1/=ARG2;
	}
	else
		ARG1=0.0;
	char result_st[20];
	sprintf(result_st,"%f",ARG1);
	result->addParam(RETURN_NAME,RETURN_TYPE,result_st);
	}
```

รูปภาพ 5-5 การเขียนเซอร์วิสโค้ดในส่วนของฟังก์ชั่น run

จากการทดลองเขียนโค้ดในส่วนดังกล่าว สามารถทำได้ง่ายโดยสามารถมองในลักษณะของการเขียนโปรแกรมเป็นฟังก์ชั่น ที่มีการเซ็ตอัพพารามิเตอร์ให้กับฟังก์ชั่นด้วยฟังก์ชั่น init ก่อน แล้วจากนั้นจึงสั่งให้ฟังก์ชั่นทำการประมวลผลโดยฟังก์ชั่น run เพื่อได้ผลลัพธ์ออกมา โดยไม่ต้องสนใจในส่วนที่ทำหน้าที่ติดต่อสื่อสารบนระบบเว็บเซอร์วิส ซึ่งซ้อนอยู่เบื้องหลัง

- เพิ่มเติมข้อมูลของการให้บริการบนระบบเว็บเซอร์วิส ลงในไฟล์ที่เก็บค่าข้อมูลดังกล่าว ที่มีชื่อว่า boot.xml ซึ่งจะหน้าที่ในการสร้างบริการขึ้นมาในระบบเมื่อระบบมีการเปิดใช้งานขึ้น โดยไฟล์ดังกล่าว เก็บอยู่ในรูปแบบของเอ็กซ์เอ็มแอล ซึ่งมีรูปแบบที่ง่ายต่อการเพิ่มเติมข้อมูลดังกล่าว โดยได้ทำการกำหนดรูปแบบที่ใช้ในการเพิ่มเติมข้อมูลไว้แล้ว ซึ่งตัวอย่างการเพิ่มข้อมูลบริการ เป็นดังรูปต่อไปนี้

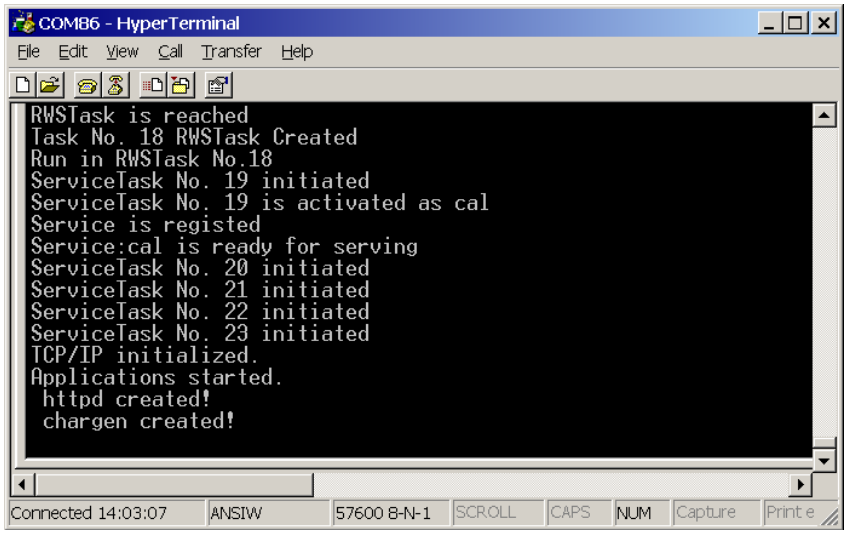
```
<cal type="key">
<Name type="string">Calculator</Name>
<Description type="string">Sample service as calculator</Description>
<Status type="string">on</Status>
<ServiceArg1 type="xsd:double">arg1</ServiceArgs1>
<ServiceArg2 type="xsd:double">arg2</ServiceArgs2>
<ServiceReturn type="xsd:double">return</ServiceReturn>
</cal>
```

รูปภาพ 5-6 แสดงการเพิ่มข้อมูลของบริการบนระบบ

จากการทดลองเพิ่มข้อมูลของบริการที่ได้ทำการพัฒนาขึ้นในระบบนั้น สามารถทำได้ง่าย เพราะไฟล์อยู่ในรูปแบบของเอ๊กซ์เอ็มแอล ซึ่งเป็นรูปแบบเอกสารที่มีลักษณะเฉพาะและเป็นมาตรฐาน หากมีตัวอย่างโครงสร้างให้แล้ว ก็สามารถทำการเพิ่มข้อมูลได้อย่างง่ายดาย

5.2.2 ทดสอบการใช้งานบริการที่พัฒนาขึ้นบนระบบ

หลังจากทำการทดลองพัฒนาบริการบนระบบแล้ว จึงสามารถทำการทดสอบการใช้งานบริการดังกล่าวบนระบบจริง โดยทำการสร้างโปรแกรมในฝั่งผู้ใช้ โดยใช้ภาษาจาวา (Java) หลังจากพัฒนาแล้วจึงทำการคอมไพล์โปรแกรมระบบเว็บเซอร์วิสบนระบบฝังตัว แล้วนำไฟล์ที่ได้ลงไปไว้บนอุปกรณ์คอม 86 แล้วทำการเริ่มการทำงานของระบบโดยการสั่งการผ่านคอนโซลซีเรียลพอร์ต



รูปภาพ 5-7 การสั่งเริ่มการทำงานของระบบเว็บเซอร์วิสบนอุปกรณ์คอม86

หลังจากให้ระบบเริ่มทำงานแล้ว ให้ทำการเรียกโปรแกรมที่ขอเรียกใช้บริการดังกล่าว ในตัวอย่างนี้ เป็นโปรแกรมที่เรียกใช้บริการ Calculator ที่ได้ทำพัฒนาในฝั่งของผู้ให้บริการจากการทดลองที่ผ่านมา โดยหลังจากเรียกใช้โปรแกรมแล้ว จะเป็นดังรูปต่อไปนี้

รูปภาพ 5-8 โปรแกรมขอเรียกใช้บริการ Calculator

แล้วทำการคลิ๊กที่ปุ่มตัวเลขที่ต้องการคำนวณ คล้ายเครื่องคิดเลข ตัวอย่าง เช่น 1 + 3 ทำได้โดยกดหมายเลข '1' ตามด้วย '+' แล้วตามด้วย '3' แล้ว '+' อีกครั้ง จากนั้น โปรแกรมจะทำการส่งข้อมูลเหล่านี้ไปขอใช้บริการ Calculator บนระบบเว็บเซอร์วิสบนอุปกรณ์คอม86 โดยหลังจากผ่านการประมวลผลแล้ว ผลที่ได้รับกลับมาจะแสดงเป็นผลลัพธ์ดังรูปต่อไปนี้

รูปภาพ 5-9 ผลลัพธ์จากการคำนวณโดยการขอใช้บริการ Calculator ผ่านเว็บเซอร์วิส