

## บทที่ 3

### วิธีการดำเนินงาน

ในบทที่ 2 ได้กล่าวถึงความรู้พื้นฐานที่จำเป็นสำหรับการบริหารจัดการเกี่ยวกับ Web Server ต่อไปจะกล่าวถึงวิธีการติดตั้งและการปรับแต่งค่าของโปรแกรม Apache Web Server เพื่อให้ Server ที่ใช้ระบบปฏิบัติการ UNIX สามารถให้บริการ Web Server ได้

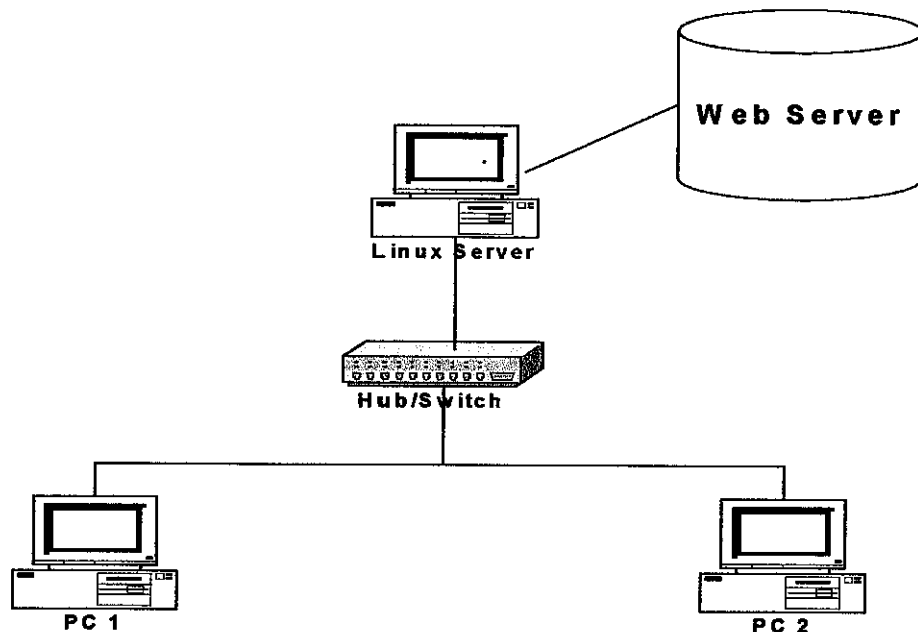
#### 3.1 วางแผนขั้นตอนก่อนการติดตั้งและการจัดการโปรแกรม Apache Web Server

##### ขั้นตอนการดำเนินงาน

1. ศึกษาหลักการทำงานของ Web Server และโปรแกรม Apache Web Server
2. วางแผนขั้นตอนการจัดการและติดตั้ง Web Server โดยใช้โปรแกรม Apache Web Server กับระบบปฏิบัติการ Linux RedHat 7.2
3. ทำการทดลองติดตั้งกับระบบปฏิบัติการ Linux RedHat 7.2 โดยใช้โปรแกรม Apache Web Server version 1.3.9
4. วางแผนขั้นตอนการจัดการและติดตั้ง Web Server โดยใช้โปรแกรม Apache Web Server กับระบบปฏิบัติการ Solaris 8.0
5. ทำการติดตั้งและการปรับแต่งค่าของโปรแกรม Apache Web Server version 1.3.9 บนระบบปฏิบัติการ Solaris 8.0
6. ทำการทดลองติดตั้งโปรแกรม Apache Web Server version 1.3.9 กับระบบที่จำลองขึ้น
7. การบริหารจัดการในส่วน Client กับ Server ของตัวโปรแกรม Apache Web Server version 1.3.9 และในส่วนของผู้ใช้กับ Administrator บนระบบปฏิบัติการ Solaris 8.

ดาวน์โหลดโปรแกรมต่างๆที่มีอยู่ทั่วโลก สามารถดูรายชื่อเว็บไซต์เหล่านี้ได้จากเว็บไซต์ <http://www.apache.org/dyn/closer>. สิ่งที่ต้องทำก่อนการติดตั้งก็คือต้องหาคู่มือการติดตั้งและใช้งานก่อน ซึ่งสามารถดาวน์โหลดได้เว็บไซต์ <http://www.apache.org/dist/httpd/apache-1.3.23-docs.pdf.zip> เป็นไฟล์ที่อยู่ในรูปแบบของ PDF ขนาดประมาณ 4 เมกกะไบต์ สำหรับผู้ใช้งานที่จะดาวน์โหลดไปติดตั้งกับระบบปฏิบัติการ Solaris ให้ดาวน์โหลดเส้นทางเดิน (path) ไปติดตั้งเพิ่มเติมด้วยโดยดาวน์โหลดที่ [http://www.apache.org/dist/httpd/patches/apply\\_to\\_1.3.23/solaris-apache-1.3.23.patch](http://www.apache.org/dist/httpd/patches/apply_to_1.3.23/solaris-apache-1.3.23.patch)

2. วางแผนขั้นตอนการจัดการและติดตั้ง Web Server โดยใช้โปรแกรม Apache Web Server กับระบบปฏิบัติการ Linux RedHat 7.2



รูปที่ 3.3 แบบจำลองระบบ Linux ที่ทำการทดลอง

การติดตั้งโปรแกรม Apache กับระบบปฏิบัติการ Linux RedHat 7.2 นี้ จะใช้โปรแกรม Apache เวอร์ชันที่ใกล้เคียงกับเวอร์ชันบนเครื่อง Server ที่ใช้ระบบปฏิบัติการ Solaris 8. เพื่อศึกษาถึงความเข้ากันได้ของระบบ ประสิทธิภาพของการให้บริการ Web Server และป้องกันไม่ให้เกิดการผิด

พลาดในการติดตั้งในระบบจริง รวมถึงป้องกันสิ่งที่เป็น Bug ของโปรแกรมเพื่อลดปัญหาการทำงาน ของ Web Server โดยการติดตั้งนี้จะมีการจำลองระบบเครือข่าย โดยทำการจัดตั้งให้มี Server ที่ใช้ ระบบปฏิบัติการ Linux RedHat 7.2 จำนวน 1 เครื่อง และมีเครื่อง Client จำนวน 2 เครื่องเชื่อมโยงเป็น เครือข่ายโดยผ่าน HUB 1 ตัว

### 3. ทำการทดลองติดตั้งโปรแกรม Apache Web Server กับระบบปฏิบัติการ Linux RedHat 7.2

การติดตั้ง Apache Web Server ใน version ที่ใกล้เคียงกับระบบปฏิบัติการ Solaris 8. บนระบบ ปฏิบัติการ Linux RedHat 7.2 ตัวอย่างการติดตั้งนี้จะใช้ Distribution ของ RedHat 7.2 โดยทำการตรวจสอบว่า ได้ทำการติดตั้ง Apache package หรือไม่ ด้วยคำสั่ง

```
# rpm -qa|grep apache
apache-1.3.3-1
```

ถ้าพบข้อความตามตัวอย่างข้างบน แสดงว่าติดตั้งเรียบร้อยแล้ว แต่ถ้ายังไม่ได้ติดตั้ง ก็ทำการติดตั้งโดย ไล่แผ่น CD Linux RedHat 7.2 แล้วทำตามคำสั่งข้างล่างนี้

```
# mount -t iso9660 /dev/cdrom /mnt
mount: block device /dev/cdrom is write-protected, mounting read-only
# cd /mnt/RedHat/RPMS
# rpm -i apache-1.3.3-1.i386.rpm
```

จากจุดนี้ ก็สามารถ สั่ง Stop - Start โปรแกรม Apache Web Server ได้ กรณีที่ใช้ x-window ให้เข้าไป ที่ Administration -> control panel-> ICON รูปคล้ายๆ ไฟเขียวไฟแดงของจราจร -> httpd -> execute -> stop or start กรณีที่ตอนติดตั้งให้ Start Web Server ตอน Boot จะไม่ต้องทำอะไรเพราะว่าโปรแกรม Apache จะถูก Start โดยอัตโนมัติ ซึ่งจะใช้ Script ที่ชื่อ /etc/rc.d/rc3.d/S85httpd ลองเข้าไปดูข้อมูลใน Script นี้จะพบข้อมูลเกี่ยวกับการปรับแต่งค่าที่สำคัญๆ ตามตัวอย่างข้างล่าง

```

# vi /etc/rc.d/rc3.d/S85httpd
#!/bin/sh

# Startup script for the Apache Web Server
# chkconfig: 345 85 15
# description: Apache is a World Wide Web server. It is used to serve \
# HTML files and CGI.
# processname: httpd
# pidfile: /var/run/httpd.pid
# config: /etc/httpd/conf/access.conf
# config: /etc/httpd/conf/httpd.conf
# config: /etc/httpd/conf/srm.conf

```

เราสามารถใช้ Script ตัวนี้ทำการควบคุม Web Server ได้ด้วยซึ่งทำให้เราไม่จำเป็นต้องใช้ x-window โดยสามารถดู Option ของ Script ด้วยคำสั่ง

```

# /etc/rc.d/rc3.d/S85httpd
Usage: /etc/rc.d/rc3.d/S85httpd {start|stop|restart|reload|status}

```

ยกตัวอย่าง กรณีที่ทำการแก้ไขการปรับแต่งค่าของโปรแกรม Apache แล้วจะ Restart จะใช้คำสั่ง

```

# /etc/rc.d/rc3.d/S85httpd restart
Shutting down http: httpd
Starting httpd: httpd

```

สำหรับ RedHat 7.2 ไม่จำเป็นต้องแก้ไขหรือปรับแต่งใดๆ ก็สามารถใช้งานได้เลย โดยป้อนข้อมูล IP Address ของ Linux ที่ Browser ดังนี้

```
http://192.168.45.10/
```

จากตัวอย่างเป็น IP ของ Linux ของเครื่องที่ทำการทดสอบ

กรณีที่ต้องการดูว่ามี process ที่ทำงานอยู่หรือไม่ให้ใช้คำสั่ง

```
# ps aux|more
```

USER	PID	%CPU	%MEM	SIZE	RSS	TTY	STAT	START	TIME	COMMAND
bin	186	0.0	0.2	752	320	?	S	06:05	0:00	portmap
nobody	827	0.0	0.8	1988	1224	?	S	11:55	0:00	httpd
nobody	828	0.0	0.8	1988	1224	?	S	11:55	0:00	httpd
nobody	829	0.0	0.8	1988	1224	?	S	11:55	0:00	httpd
nobody	830	0.0	0.8	1988	1224	?	S	11:55	0:00	httpd
nobody	831	0.0	0.8	1988	1224	?	S	11:55	0:00	httpd
nobody	832	0.0	0.8	1988	1224	?	S	11:55	0:00	httpd
nobody	833	0.0	0.8	1988	1224	?	S	11:55	0:00	httpd
nobody	834	0.0	0.8	1988	1224	?	S	11:55	0:00	httpd
nobody	835	0.0	0.8	1988	1224	?	S	11:55	0:00	httpd
nobody	836	0.0	0.8	1988	1224	?	S	11:55	0:00	httpd
root	1	0.0	0.2	764	388	?	S	06:05	0:03	init

ข้างบนนี้เป็นตัวอย่าง process ที่ทำงานอยู่จากเครื่องที่ทดสอบ โดย httpd ก็จะเป็น process ของโปรแกรม Apache Web Server และ ถ้าอยากดูการ Access Web Page หรือ Error Log ให้ดู Logfile ที่

```
/var/log/httpd/access_log
```

```
/var/log/httpd/error_log
```

และถ้าจะแก้ไข Home Page ก็ให้ไปแก้ที่ /home/httpd/html/index.html โดยทดลองใช้คำสั่ง

```
# cd /home/httpd/html
```

```
# mv index.html index.html.org
```

```
# vi index.html
```

โดยให้ข้อมูลใน File index.html มีข้อมูลตามตัวอย่างข้างล่าง

```
<html>
    test
</html>
```

แล้วทำการลองเรียก File index.html ที่ Browser จะได้เป็น File index.html ที่ สร้างขึ้นมาใหม่  
ในกรณีที่ต้องการแก้ไข ไคเร็กทอรีที่เก็บ homepage จากค่าเดิมที่อยู่ที่ /home/httpd/html ให้ไปแก้ที่ File  
/etc/httpd/conf/srm.conf โดยต้องแก้ตรงบรรทัดตามตัวอย่างดังนี้

```
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#DocumentRoot /home/httpd/html
#DocumentRoot /home/httpd/newhtml
```

และต้องไปสร้าง ไคเร็กทอรี ให้ตรงกับที่ปรับแต่งค่าไว้และทำการ Restart โปรแกรม Apache  
Web Server

#### 4. วางแผนขั้นตอนการจัดการและติดตั้ง Web Server โดยใช้โปรแกรม Apache Web Server กับ ระบบปฏิบัติการ Solaris 8.0

หลังจากทำการทดลอง การติดตั้ง Apache Web Server กับระบบปฏิบัติการ Linux เพื่อทดสอบ  
ระบบและการให้บริการ Web Server ก่อนนำไปติดตั้งจริงในระบบปฏิบัติการ Solaris 8. ซึ่งได้ผลเป็น  
ที่น่าพอใจแล้ว ต่อจากนั้นได้ทำการวางแผนขั้นตอนการติดตั้งและการบริหารจัดการ Web Server บน  
ระบบปฏิบัติการ Solaris 8 ไว้ดังนี้

1. ทำการศึกษาว่าโปรแกรม Apache Web Server สามารถติดตั้งกับระบบปฏิบัติการ Solaris 8 ได้  
หรือไม่โดยศึกษาจากคู่มือของระบบปฏิบัติการ Solaris 8 และ Web Site ของโปรแกรม

Apache Web Server สรุปได้ว่าโปรแกรม Apache Web server สามารถติดตั้งกับระบบปฏิบัติการ Solaris 8 ได้ โดยตัว โปรแกรม Apache Web server เป็นส่วนหนึ่งใน Package ของระบบปฏิบัติการ Solaris 8

2. ทำการจัดหาตัวโปรแกรม Apache Web Server version ที่สามารถติดตั้งกับระบบปฏิบัติการ Solaris 8 จาก Web Site ของโปรแกรม Apache Web Server และ Package Software ของระบบปฏิบัติการ Solaris 8
  3. ศึกษาคู่มือการติดตั้งและการ ปรับแต่งค่าของโปรแกรม Apache Web Server
  4. จัดเตรียมระบบเครือข่ายที่จะทำการติดตั้ง ทำการตรวจสอบเครื่อง UNIX ที่จัดตั้งให้เป็น Server และเครื่อง Client ว่าสามารถเชื่อมโยงกันเป็นระบบเครือข่ายเดียวกันหรือไม่
5. ทำการติดตั้งและปรับแต่งค่าของโปรแกรม Apache Web Server version 1.3.23 บนระบบปฏิบัติการ solaris 8.0

ขั้นตอนการติดตั้งโปรแกรม Apache Web Server version 1.3.23

ขั้นตอนการติดตั้งโปรแกรม Apache โดยสรุปจะมีดังนี้

1. ขยายไฟล์
2. แยกไฟล์เป็นไดเรกทอรีย่อย
3. คอมไพล์โปรแกรม Apache
4. ติดตั้งโปรแกรม Apache
5. ทดสอบรันโปรแกรม Apache Web Server version 1.3.9
6. ปรับแต่งโปรแกรม Apache

ขั้นตอนที่ 1 ขยายไฟล์

ก่อนอื่นให้ก๊อปปี้โปรแกรมที่ดาวน์โหลดมารวบรวมไว้ที่เดียวกันเพื่อความเป็นระเบียบเรียบร้อย โดยจะสร้างไดเรกทอรี /usr/local/src ไว้สำหรับเก็บซอสโค้ดของโปรแกรมต่าง ๆ ที่ดาวน์โหลดมาติดตั้ง แล้วทำตามขั้นตอนดังนี้

1. ย้ายไดเรกทอรีมายังไดเรกทอรีที่เก็บซอสโค้ดของโปรแกรม Apache ซึ่งในที่นี้ก็คือ /usr/local/src โดยใช้คำสั่ง

```
#cp xxx /usr/local/src
```

2. ใช้คำสั่ง ls เพื่อเรียกดูรายชื่อไฟล์
3. แสดงรายชื่อไฟล์ apache-1.3.23.tar.gz ซึ่งเป็นไฟล์ที่ถูกบีบอัดข้อมูลไว้
4. ใช้คำสั่ง gunzip เพื่อขยายไฟล์ที่ถูกบีบอัดไว้ออกมาใช้งาน
5. ใช้คำสั่ง ls เพื่อเรียกดูรายชื่อไฟล์
6. แสดงรายชื่อไฟล์ apache-1.3.23.tar จะสังเกตว่านามสกุล .gz นั้นหายไปแล้ว กลายเป็นไฟล์ธรรมดาที่ไม่ได้ถูกบีบอัดข้อมูล

ในระบบปฏิบัติการยูนิกซ์จะมีการบีบอัดข้อมูลหลายประเภทแต่ที่นิยมกันมากที่สุดก็คือบีบอัดในอยู่ในรูป .gz โดยการใช้โปรแกรม gzip สำหรับบีบอัดไฟล์ซึ่งคล้าย กับโปรแกรม WinZip ของวินโดวส์ ถ้าเรามีไฟล์ ชื่อ data.txt จะใช้คำสั่ง "gzip data.txt" ไฟล์ data.txt จะถูกบีบอัดข้อมูล และเปลี่ยนชื่อใหม่เป็น data.txt.gz คือชื่อไฟล์เหมือนเดิมแต่ต่อท้ายด้วย .gz จะสังเกตได้ว่าชื่อไฟล์ในระบบปฏิบัติการยูนิกซ์นั้นมีจุดได้หลายจุด เราจะถือว่าจุดท้ายสุดคือนามสกุลของไฟล์นั้น ๆ ตัวอย่างเช่นในที่นี้ไฟล์ชื่อ data.txt.gz ก็จะถือว่าเป็นไฟล์ที่มีนามสกุลเป็น .gz

## ขั้นตอนที่ 2 แยกไฟล์เป็นไดเรกทอรี

หลังจากที่ขยายไฟล์ออกมาจนได้ไฟล์ Apache-1.3.23.tar แล้ว เราก็จะแยกไฟล์ที่มีนามสกุลเป็น .tar นี้ให้เป็นไดเรกทอรี โดยใช้คำสั่ง tar ซึ่งมีรายละเอียดในขั้นตอนี่ดังนี้

1. ใช้คำสั่ง "tar xvf apache-1.3.23.tar" เพื่อแยกไฟล์ให้เป็นไดเรกทอรี
2. ใช้คำสั่ง ls เพื่อตรวจดูรายชื่อไฟล์ในไดเรกทอรี
3. แสดงรายชื่อไฟล์ไดเรกทอรีสองชื่อคือ apache-1.3.23 และ ไฟล์ apache-1.3.23.tar ซึ่งชื่อ apache-1.3.23 ก็คือไดเรกทอรีที่แตกออกมาจากไฟล์ apache-1.3.23 ไฟล์ที่มีนามสกุลเป็น .tar คือไฟล์ที่เกิดจากการรวมไฟล์หรือไดเรกทอรีที่มีไฟล์ มีไดเรกทอรีย่อย ๆ อยู่ภายใน ให้รวมเป็นไฟล์เดียว เพื่อสะดวกต่อการจัดเก็บ แต่การรวมไดเรกทอรีให้เป็นไฟล์เดียวนี้อาจไม่มีการบีบอัดข้อมูล จะรวมไดเรกทอรีเป็นไฟล์ จากนั้นจึงจะใช้โปรแกรม gzip บีบอัดข้อมูลอีกครั้ง

ตัวอย่างเช่น ไดเรกทอรี data มีไฟล์ย่อยอยู่ตามไฟล์คือ data1.txt, data2.txt, data3.txt เมื่อเรารวมไดเรกทอรีให้เป็นไฟล์ด้วยคำสั่ง "tar cvf data.tar data" จะหมายถึงการรวมไฟล์ย่อยทั้งหมดที่อยู่ในไดเรกทอรี data ให้เป็นไฟล์เดียวชื่อ data.tar ซึ่งไฟล์นี้จะเก็บไฟล์ data1.txt, data2.txt และ



data3.txt ไว้ภายในด้วย ซึ่งไฟล์ apache-1.3.23.tar ก็เช่นเดียวกันจะมีไฟล์ที่รวมไคลเอนท์และไฟล์ไว้ จึงต้องแตกออกมาก่อน

### ขั้นตอนที่ 3 การคอมไพล์โปรแกรม Apache Web Server version 1.3.9

โปรแกรมที่เราแตกออกมาเป็นไคลเอนท์แล้วนั้นยังเป็นโปรแกรมที่ไม่สามารถใช้งานได้ เนื่องจากเป็นซอร์สโค้ดเราจะต้องนำซอร์สโค้ดมาคอมไพล์ให้เป็นโปรแกรมสำหรับใช้งานก่อน สาเหตุที่เป็นดังนี้ก็เพราะว่า ผู้พัฒนาต้องการให้เกิดความยืดหยุ่นในการแจกจ่ายซอฟต์แวร์ คือเขียนครั้งเดียวให้สามารถใช้งานได้กับระบบปฏิบัติการหลาย ๆ ระบบ โดยโปรแกรมประเภทโอเพนซอร์สนี้จะพัฒนาด้วยภาษาที่เป็นกลาง เช่น ภาษาซี แล้วแจกซอร์สโค้ดให้ผู้ใช้งานนำไปคอมไพล์บนเครื่องของตนเองด้วยตัวแปลภาษาของเครื่องนั้น ๆ ด้วยเหตุนี้โปรแกรมที่แจกจ่ายซอร์สโค้ดจึงสามารถทำงานได้บนทุกระบบปฏิบัติการที่มีตัวแปลภาษาแบบเดียวกับภาษาที่ใช้พัฒนา สำหรับโปรแกรม Apache นั้นสามารถคอมไพล์ได้บนระบบปฏิบัติการดังต่อไปนี้

ระบบปฏิบัติการที่สามารถคอมไพล์โปรแกรม Apache Web Server version 1.3.9 ได้

1. Linux
2. FreeBSD
3. OpenBSD
4. NetBSD
5. Digital Unix
6. SunOS
7. Solaris
8. IRIX
9. HPUX
10. DGUX
11. UnixWare
12. AIX
13. SCO
14. ReliantUNIX
15. Darwin/Mac OS

16. OpenStep/Mach

17. DYNIX/ptx

18. BSDI

นอกจากระบบปฏิบัติการที่สามารถคอมไพล์โปรแกรม Apache ได้แล้วก็จะมีใบนารีไฟล์ของระบบปฏิบัติการต่างๆ ที่คอมไพล์ไว้ให้แล้วอีกด้วยสามารถนำไปติดตั้งได้ทันทีโดยไม่ต้องคอมไพล์อีกอย่างเช่นระบบปฏิบัติการวินโดวส์

**ขั้นตอนในการคอมไพล์หรือแปลภาษาของโปรแกรม Apache Web Server version 1.3.9**

1. คำสั่งปรับแต่งค่าเป็นคำสั่งที่ใช้ตรวจสอบสถานะแวดล้อมของระบบปฏิบัติการและกำหนดเงื่อนไขในการคอมไพล์โปรแกรม
2. เป็นบางส่วนหน้าจอของคำสั่งปรับแต่งค่าขณะที่ทำงาน ข้อความที่ปรากฏบนหน้าจอจะต้องไม่แสดงข้อผิดพลาด
3. ป้อนคำสั่ง make ซึ่งเป็นคำสั่งที่ใช้ในการคอมไพล์โปรแกรม ต้องระวังไม่ให้มีข้อผิดพลาดเกิดขึ้น

#### ชนิดของข้อผิดพลาด

การทำความเข้าใจเกี่ยวกับชนิดของข้อผิดพลาดระหว่างการคอมไพล์โปรแกรม ระหว่างการคอมไพล์โปรแกรมนั้นจะมีข้อผิดพลาดเกิดขึ้นสองอย่างคือ Warning และ Error

**Warning** เป็นข้อผิดพลาดเล็กน้อยของโปรแกรมเช่นมีการใช้พารามิเตอร์ในการคอมไพล์ที่ไม่จำเป็น หรือมีการประกาศตัวแปรไว้ซ้ำกัน หรือมีการประกาศตัวแปรบางชนิดไว้แล้วไม่มีการใช้งาน ข้อผิดพลาดในลักษณะนี้สามารถมองข้ามได้ ซึ่งโปรแกรมจะแสดงข้อผิดพลาดเหล่านี้เป็นการเตือน และทำการคอมไพล์ต่อไปจนจบกระบวนการ ข้อความเตือนนี้จะเกิดขึ้นเป็นประจำสำหรับการคอมไพล์ซอฟต์แวร์โอเพนซอร์สเนื่องจากตัวแปลภาษาบนแต่ละระบบปฏิบัติการจะไม่เหมือนกันทั้งหมด จึงมีข้อผิดพลาดบ้างแต่ไม่ส่งผลกระทบต่อการทำงาน

**Error** เป็นข้อผิดพลาดขั้นรุนแรงที่ไม่สามารถคอมไพล์โปรแกรมได้ เมื่อเกิดข้อผิดพลาดในลักษณะนี้เครื่องจะหยุดการคอมไพล์ทันที เหตุการณ์ Error เหล่านี้ได้แก่ มีการเรียกใช้ฟังก์ชันที่ไม่มีอยู่ในคอมไพล์เลอร์ ไม่สามารถคอมไพล์ตามพารามิเตอร์ที่ป้อนได้ ตัวแปลภาษาไม่รู้จักคำสั่งที่กำหนด ความผิดพลาดเหล่านี้ระบบปฏิบัติการจะไม่สามารถทำงานต่อไปได้ โดยส่วนใหญ่แล้วมัก

จะเกิดเมื่อนำซอร์สโค้ดไปคอมไพล์บนระบบปฏิบัติการอื่นหรือคอมไพล์ด้วยตัวแปลภาษาที่ผิดจากที่ผู้สร้างโปรแกรมได้แนะนำไว้

### ความต้องการพื้นฐานเพื่อคอมไพล์โปรแกรม เนื้อหาของฮาร์ดดิสก์

โปรแกรม Apache จะใช้เนื้อที่ในการคอมไพล์ประมาณ 12 เมกกะไบต์สำหรับการทำงาน และหลังจากคอมไพล์เรียบร้อยแล้วจะใช้เนื้อหาของฮาร์ดดิสก์ ประมาณ 3 เมกกะไบต์เท่านั้น ซึ่งเนื้อที่บนฮาร์ดดิสก์นี้จะขึ้นอยู่กับออปชันของการคอมไพล์ด้วยว่าจะเลือกออปชันหรือโหลดโมดูลมากน้อยเพียงไหน จะเห็นได้ว่าใช้เนื้อที่บนฮาร์ดดิสก์น้อยกว่าซอฟต์แวร์ลิขสิทธิ์บางตัวมาก

### คอมไพล์เลอร์ภาษา C

ในการคอมไพล์โปรแกรม Apache นี้จะใช้คอมไพล์เลอร์ที่รองรับมาตรฐาน ANSI-C เช่น GNU C compiler (GCC) ของ Free Software Foundation รุ่น 2.7.2 ขึ้นไป ซึ่งสามารถดาวน์โหลดได้จากเว็บไซต์ <http://www.gnu.org> แต่ถ้าใช้คอมไพล์เลอร์ตัวอื่นที่ไม่ใช่ GCC เช่น CC ก็จะต้องรองรับมาตรฐาน ANSI-C ด้วย

### คอมไพล์เลอร์ภาษา Perl

โปรแกรม Apache สามารถทำงานกับโปรแกรมอื่น ๆ ได้ซึ่งหลายโปรแกรมมักจะต้องคอมไพล์เลอร์ภาษา Perl ด้วย แนะนำให้ติดตั้ง คอมไพล์เลอร์ภาษา Perl รุ่น 5.003 ขึ้นไป

### การกำหนดพารามิเตอร์ในการคอมไพล์แบบพิเศษ

Apache สามารถคอมไพล์ได้หลายรูปแบบมาก ซึ่งสามารถปรับแต่งการคอมไพล์ให้เหมาะสมกับสภาพการทำงานของระบบได้โดยการกำหนดออปชันในการคอมไพล์ แต่ถ้าทำการป้อนคำสั่ง `./configure` โดยไม่กำหนดออปชันหรือพารามิเตอร์อื่นเพิ่มเติม โปรแกรม `configure` จะกำหนดค่าพารามิเตอร์เหล่านี้ให้เองโดยอัตโนมัติ เช่น ไดรฟ์กทอรีเริ่มต้นของการติดตั้งโปรแกรมจะกำหนดให้เป็น `/user/local/apache` แต่ถ้าเราต้องการติดตั้งโปรแกรม `apache` ที่ไดเรกทอรีอื่นก็สามารถกำหนดออปชันเพิ่มเติมให้คำสั่ง `./configure` ได้ดังนี้

```
#!/configure --prefix=PREFIX
```

โดยที่ PREFIX คือไดเรกทอรีเริ่มต้นสำหรับการติดตั้งโปรแกรม Apache ตัวอย่างเช่นถ้าเราต้องการติดตั้งโปรแกรม Apache ในลงในไดเรกทอรี Apache แทนที่จะเป็น /usr/local/apache ก็ให้ใช้คำสั่งในการคอมไพล์ดังนี้

```
#!/configure - -prefix=/Apache
```

ในกรณีที่ต้องการเพิ่มโมดูลต่าง ๆ เข้าไปใน Apache ก็ให้เพิ่มเมื่อใช้คำสั่ง ./configure เช่น

```
./configure - -prefix=/usr/local/apache \  
- -enable-module=rewrite - -enable-shared=rewrite \  
- -enable-module=proxy - -enable-shared=proxy
```

การโหลดโมดูล คือ การกำหนดให้ Apache รองรับการทำงานพิเศษแบบต่าง ๆ แต่ถ้าต้องการโหลดฟังก์ชันการทำงานทุกอย่างให้ใช้พารามิเตอร์สำหรับคำสั่ง ./configure เป็น

```
./configure - -enable-module=all --enable-shared=max
```

การกำหนดคอปชันอย่างนี้ไม่เป็นที่นิยมเพราะขนาดของโปรแกรมที่คอมไพล์แล้วจะใหญ่มาก จะใช้เวลาโหลดนาน ควรโหลดโมดูลต่าง ๆ เท่าที่จำเป็นในการใช้งาน

โมดูลของ Apache มีอยู่เป็นจำนวนมาก มีทั้งโมดูลที่พัฒนาโดยทีมงานของ Apache เองและโมดูลที่พัฒนาโดยบุคคลอื่นเราสามารถตรวจสอบรายชื่อโมดูลต่างๆ ได้ที่ <http://www.apache.org/dist/httpd/contrib/modules/> การโหลดโมดูลนี้จะช่วยให้สามารถทำงานได้สะดวกขึ้น

#### ขั้นตอนที่ 4 ติดตั้งโปรแกรม Apache Web Server version 1.3.9

โปรแกรม Apache จะทำการติดตั้งโปรแกรมที่คอมไพล์แล้ว สร้างไฟล์และไดเรกทอรีที่เกี่ยวข้องให้เรียบร้อยด้วยคำสั่ง `make install`

#### ขั้นตอนที่ 5 ทดสอบรันโปรแกรม Apache Web Server version 1.3.9

Apache จะมีโปรแกรมย่อยหลายตัวที่ช่วยในการทำงานด้วย ดังต่อไปนี้

**httpd**

Apache hypertext transfer protocol server : เป็นโปรแกรมสำหรับทำหน้าที่เป็นเว็บเซิร์ฟเวอร์

**apachectl**

Apache HTTP server control interface : เป็นโปรแกรมสำหรับควบคุมการทำงานของเว็บเซิร์ฟเวอร์เช่น เริ่มต้นการทำงาน, หยุดการทำงาน ,ทดสอบการปรับแต่งค่าไฟล์ต่างๆ

**apxs**

Apache extension tool : เป็นโปรแกรมเสริมสำหรับ Apache

**dbmmanage**

Create and update user authentication files in DBM format for basic authentication : เป็นโปรแกรมสำหรับจัดการเกี่ยวกับสิทธิ์ ของผู้ใช้งานในรูปแบบ DBM

**htdigest**

Create and update user authentication files for digest authentication : เป็นโปรแกรมสำหรับการจัดการสิทธิ์สำหรับ Digest Authentication

**htpasswd**

Create and update user authentication files for basic authentication : เป็นโปรแกรมสำหรับการจัดการสิทธิ์ของผู้ใช้งานขั้นพื้นฐานในรูปแบบรหัสผ่านของยูนิคส์

**logresolve**

Resolve hostnames for IP-addresses in Apache logfiles : โปรแกรมสำหรับการจัดการระบบโดเมนเนมในล็อกไฟล์ของโปรแกรม Apache

**rotatelog**

Rotate Apache logs without having to kill the server : โปรแกรมสำหรับจัดการล็อกไฟล์ไม่ให้อุปกรณ์การทำงานของเซิร์ฟเวอร์ การทำงานหลักของโปรแกรม Apache จะทำงานด้วยโปรแกรม httpd ซึ่งทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ สำหรับการแสดงโปรแกรม Apache เพื่อทดสอบให้ใช้คำสั่ง

```
#/usr/local/apache/bin/apachectl start
```

โปรแกรม apachectl นี้จะเป็นสคริปต์สำหรับการเรียกใช้งานโปรแกรม httpd อีกทีหนึ่ง

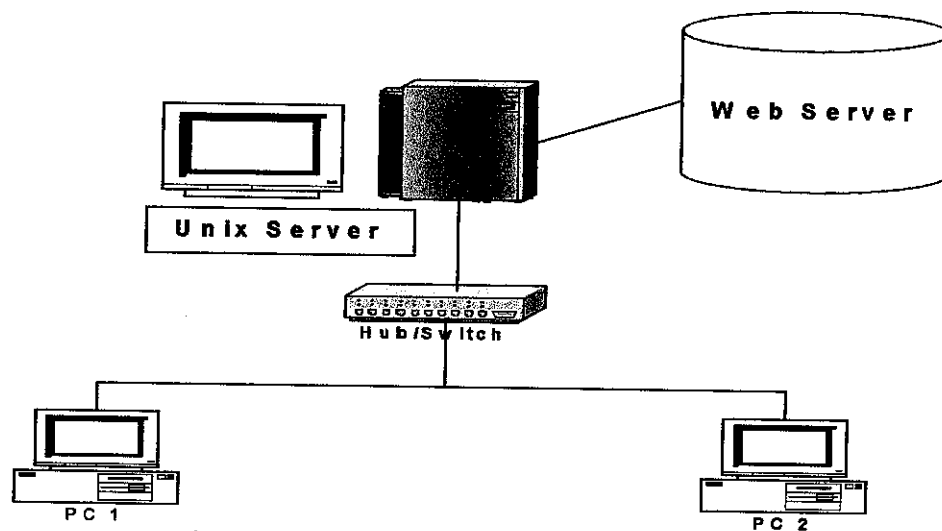
### ขั้นตอนที่ 6 ปรับแต่งโปรแกรม Apache Web Server version 1.3.9

เมื่อติดตั้งโปรแกรม Apache เรียบร้อยแล้ว Apache จะพร้อมสำหรับการใช้งานในระดับหนึ่งเท่านั้น ซึ่งเราสามารถปรับแต่งเพิ่มเติมภายหลังการคอมไพล์ได้โดยการแก้ไขไฟล์ที่เกี่ยวข้อง 3 ไฟล์ คือ

```
#/usr/local/apache/conf/httpd.conf
#/usr/local/apache/conf/srm.conf
#/usr/local/apache/conf/access.conf
```

ไฟล์ httpd.conf จะเป็นการปรับแต่งเซิร์ฟเวอร์ทั่วไปเช่น หมายเลขพอร์ตที่ใช้งาน ชื่อผู้ใช้งานที่จะทำงานบนโปรแกรมนี้ และไฟล์ srm.conf จะเป็นการปรับแต่งค่าเกี่ยวกับไคลเอนต์และฟังก์ชันพิเศษ ส่วนไฟล์ access.conf จะเป็นการจัดการเกี่ยวกับสิทธิ์ในการเรียกดูข้อมูลของ User

### 6. ทำการทดลองติดตั้ง Program Apache Web Server version 1.3.9 กับระบบ UNIX ที่จำลองขึ้น



รูปที่ 3.4 แบบจำลองของ ระบบ UNIX ที่ทำการทดลอง

### การสตาร์ทโปรแกรม Apache Web Server version 1.3.9

โปรแกรมเว็บเซิร์ฟเวอร์จะทำงานเป็น Background Process หรือทำงานอยู่เบื้องหลังอยู่ตลอดเวลา โดยโปรแกรมหลักที่ชื่อ httpd จะถูกตั้งให้ทำงานไว้เพื่อคอยควบคุมการติดต่อรับส่งข้อมูลที่

พอร์ท 80 เราสามารถเปลี่ยนแปลงได้โดยการแก้ไขที่ไฟล์ `httpd.conf` การสตา์ทโปรแกรมนั้นสามารถทำได้สองวิธีคือ วิธีที่ 1. เรียกโปรแกรม Apache โดยตรงด้วยคำสั่ง

```
#/usr/local/apache/bin/httpd -f /usr/local/apache/conf/httpd.conf
```

ฟังก์ชัน `-f` มีไว้ระบุไฟล์ที่ใช้ปรับแต่งค่าของโปรแกรม Apache ซึ่งสามารถกำหนดเป็นไฟล์อื่นได้ ส่วนวิธีที่ 2. คือการเรียกผ่าน โปรแกรม `apachectl` ดังนี้

```
#/usr/local/apache/bin/apachectl start
```

ถ้าหากการทำงานไม่มีปัญหาหรือความผิดพลาด การเรียกคำสั่งนี้จะใช้เวลาไม่กี่วินาทีแล้วก็จะแสดงผลออกมาให้ และเมื่อมีการเรียกโฮมเพจจะแสดงไฟล์ Web Page ที่อยู่ในไดเรกทอรี `/usr/local/apache/htdocs` ซึ่งเป็นไดเรกทอรีสำหรับเก็บโฮมเพจ

### ปัญหาในการสตา์ทโปรแกรม Apache Web Server version 1.3.9

ถ้ามีข้อผิดพลาดในการสตา์ทโปรแกรม Apache เกิดขึ้นสามารถดูรายละเอียดความผิดพลาดในการสตา์ทโปรแกรมได้ที่ไฟล์ `ErrorLog` สำหรับข้อผิดพลาดที่พบบ่อยก็คือ

1. สตา์ทด้วย User ที่ไม่มีสิทธิ์ในการสั่งทำงานโปรแกรม
2. สตา์ทโปรแกรม Apache ซ้ำหรือ มีโปรแกรมอื่นใช้งานที่พอร์ทชนกัน

### การสตา์ทโปรแกรม Apache Web Server version 1.3.9 อัตโนมัติเมื่อเปิดเครื่อง

ถ้าต้องการให้โปรแกรม Apache สตา์ทอัตโนมัติเวลาเปิดเครื่องใหม่ ก็ให้ทำการป้อนคำสั่งในการสตา์ทโปรแกรม Apache ไว้ที่ไฟล์ `rc.local` ในไดเรกทอรี `/etc` โดยเพิ่มคำสั่งในบรรทัดนี้เข้าไปที่ท้ายไฟล์

```
/usr/local/apache/bin/apachectl start
```

### การหยุดและรีสตาร์ทโปรแกรม Apache

เราสามารถหยุดและรีสตาร์ทโปรแกรมได้โดยการจัดการกับ Process ของโปรแกรม Apache คือยกเลิก Process httpd และรีสตาร์ท httpd ใหม่ แต่วิธีที่สะดวกที่สุดก็คือหยุดและรีสตาร์ทผ่านโปรแกรม apachectl โดยมีคำสั่งดังนี้

`/usr/local/apache/bin/apachectl stop` : หยุดโปรแกรม Apache

`usr/local/apache/bin/apachectl restart` : รีสตาร์ทโปรแกรม Apache

### การสร้างโฮมเพจ

เมื่อทำการติดตั้งและทดสอบการใช้งานโปรแกรม Apache เรียบร้อยแล้ว ให้สร้างเว็บเพจแล้วก็อปปี้อมาไว้ในไดเรกทอรี `/usr/local/apache/htdocs` ก็จะสามารถเรียกดูเว็บเพจนั้นได้

## 7. การบริหารจัดการในส่วน Client กับ Server ของตัวโปรแกรม Apache Web Server version 1.3.9 และในส่วนของ User กับ Administrator บนระบบปฏิบัติการ Solaris 8.

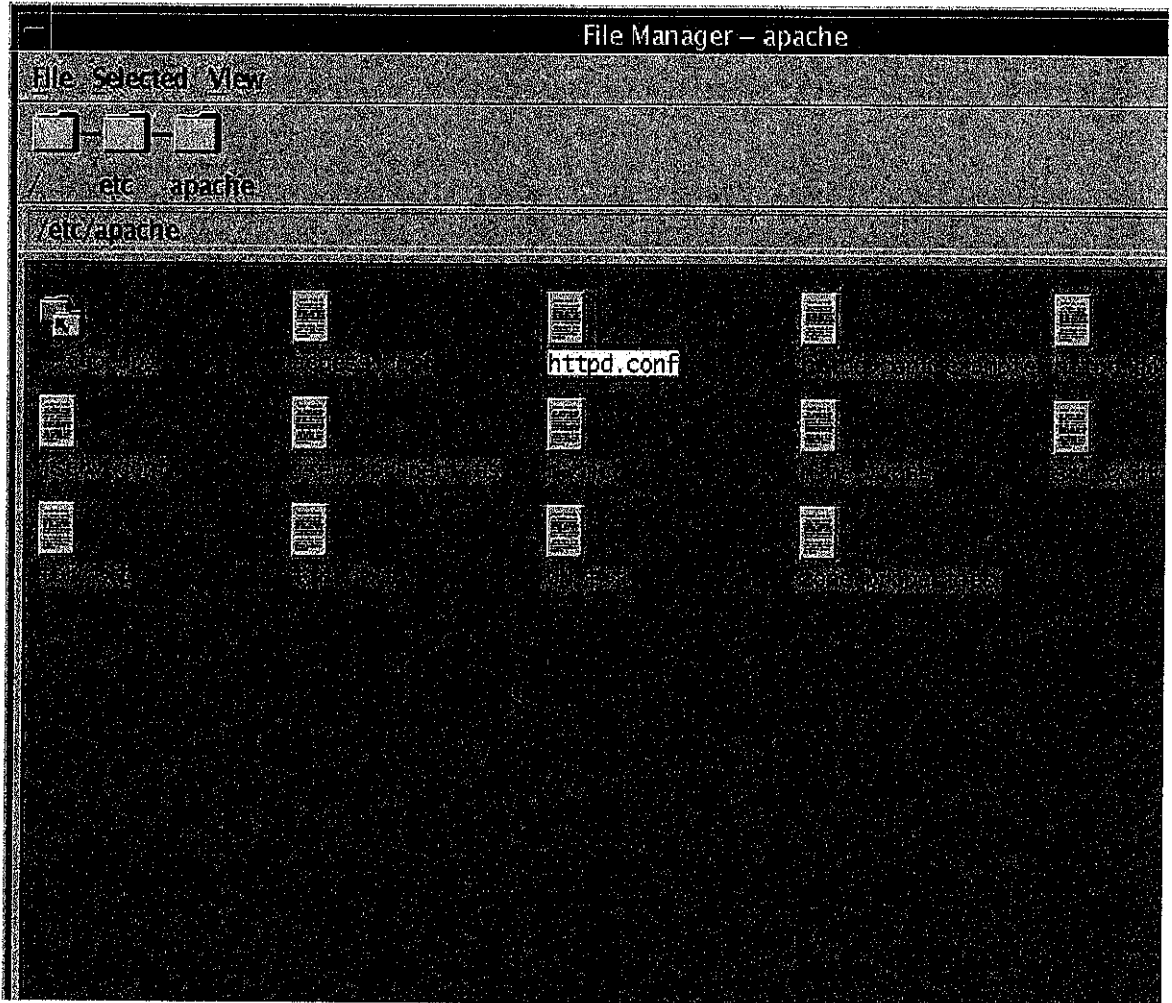
หลังจากที่ทำการลงโปรแกรม Apache Web Server และ ปรับแต่งค่า ของตัวโปรแกรมเป็นที่เรียบร้อยแล้วจะต้องทำการเกี่ยวกับการบริหารจัดการในส่วนของ Client กับ Server ของตัวโปรแกรม Apache Web Server เพื่อจัดการในส่วนของพื้นที่ในการสร้างและเก็บ Home Page ของแต่ละ User อย่างเป็นสัดส่วนและเป็นการง่ายต่อการดูแลและจัดการของ Administrator ในส่วนของ User กับ Administrator บนระบบปฏิบัติการ Solaris 8 จะทำการบริหารจัดการเกี่ยวกับการเพิ่ม User ในระบบ และการจัดการสิทธิในการเข้าถึงระบบของแต่ละ User เพื่อรักษาความปลอดภัยของระบบ

### ส่วน Client กับ Server ของตัวโปรแกรม Apache Web Server

1. การปรับแต่งโปรแกรม Apache Web Server ในส่วนนี้จะปรับแต่ง module ของการ ตั้ง ไดเรกทอรี ที่ใช้เป็น Local Host URL ในการเรียก Web Page ของแต่ละ User โดยปรับแต่งที่ ไฟล์ `httpd.conf` ดังนี้



- การเข้าไปปรับแต่งค่าไฟล์ httpd.conf ให้เข้าไปใน /etc/apache ซึ่งเป็นที่เก็บไฟล์ httpd.conf ดังรูปที่ 3.5



รูปที่ 3.5 ไดรฟ์ทอรีที่เก็บไฟล์ httpd.conf

- ไฟล์ `httpd.conf` เป็นไฟล์ที่ใช้ระบุนรายละเอียดของการปรับแต่งค่าหลาย ๆ อย่างของ Apache ให้ทำการ เปลี่ยนที่ Server Root เป็น `/var/apache` เพื่อกำหนดให้ Server รู้ว่า Root ของตัวโปรแกรมนี้อยู่ที่ไหน ดังรูปที่ 3.6

```

# different IP addresses or hostnames and have them handled by the
# same Apache server process.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path.  If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "logs/foo.log"
# with ServerRoot set to "/usr/local/apache" will be interpreted by the
# server as "/usr/local/apache/logs/foo.log".
#
### Section 1: Global Environment
#
# The directives in this section affect the overall operation of Apache,
# such as the number of concurrent requests it can handle or where it
# can find its configuration files.
#
#
# ServerType is either inetd, or standalone.  Inetd mode is only supported on
# Unix platforms.
#
ServerType standalone
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE!  If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the LockFile documentation
# (available at <URL:http://www.apache.org/docs/mod/core.html#lockfile>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
#
#
# The LockFile directive sets the path to the lockfile used when Apache
# is compiled with either USE_FCNTL_SERIALIZED_ACCEPT or
# USE_FLOCK_SERIALIZED_ACCEPT.  This directive should normally be left at
# its default value.  The main reason for changing it is if the logs
# directory is NFS mounted, since the lockfile MUST BE STORED ON A LOCAL
# DISK.  The PID of the main server process is automatically appended to
# the filename.
#

```

รูปที่ 3.6 module ในส่วนของ Server Root เพื่อการปรับแต่งค่าโปรแกรม Apache Web Server

- เปลี่ยนที่ Document Root เป็น /var/apache/htdocs เพื่อกำหนดให้ Server รู้ว่า เป็นที่เก็บ Document อยู่ตรง path นี้

```

Text Editor - httpd.conf
File Edit Format Options Help
# NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
# when the value of (unsigned)Group is above 60000;
# don't use Group #-1 on these systems!
#
User nobody
Group nobody

#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents.
#
ServerAdmin you@your.address

#
# ServerName allows you to set a host name which is sent back to clients for
# your server if it's different than the one the program would get (i.e., use
# "www" instead of the host's real name).
#
# Note: You cannot just invent host names and hope they work. The name you
# define here must be a valid DNS name for your host. If you don't understand
# this, ask your network administrator.
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address (e.g., http://123.45.67.89/)
# anyway, and this will make redirections work in a sensible way.
#
#ServerName new.host.name

ServerName SUN-EN614

#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
#
# Each directory to which Apache has access, can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very restrictive set of
# permissions.
#

```

รูปที่ 3.7 module ในส่วนของ DocumentRoot

- เปลี่ยนที่ UserDir เป็นที่ public\_html เพื่อกำหนดให้ Server รู้ว่า เป็นที่เก็บไฟล์ Web Page ของแต่ละ User จะอยู่ตรงใดเรียกทอริ นี้

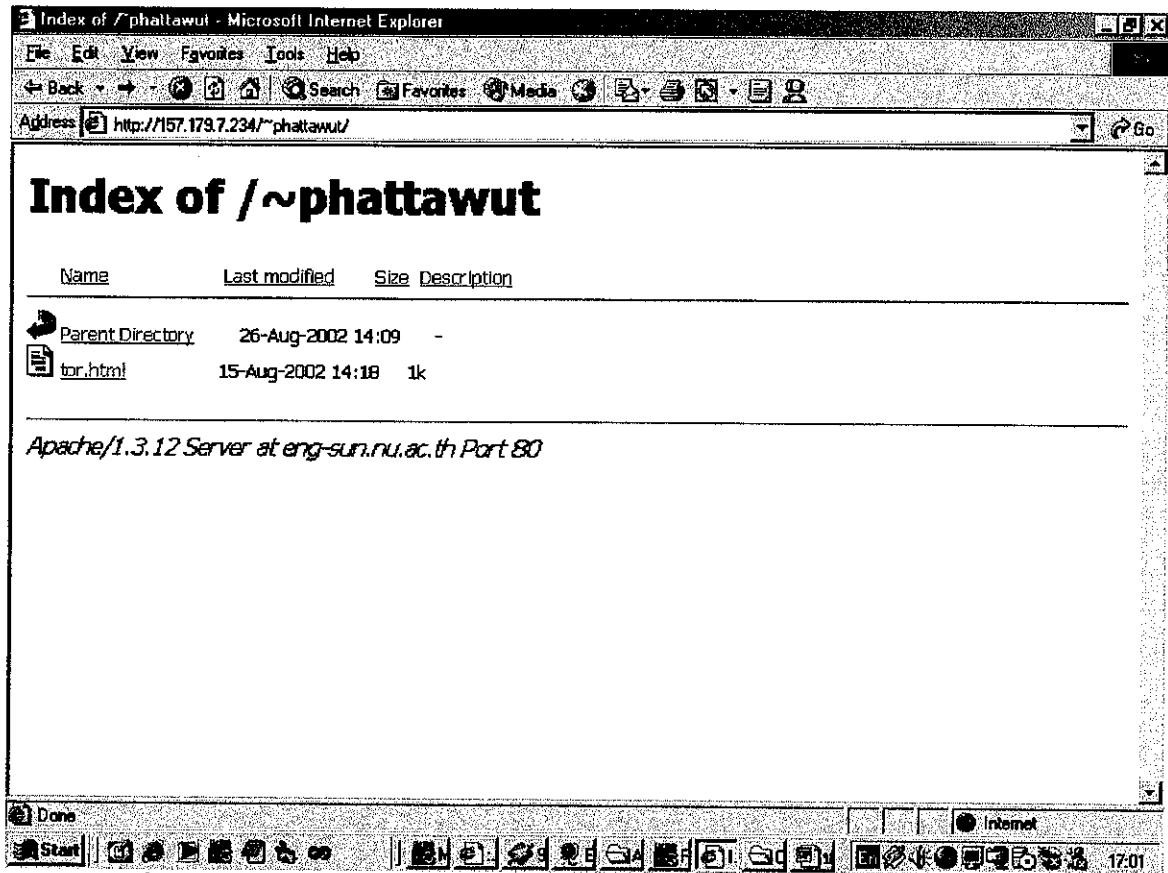
```

Text Editor - httpd.conf
File Edit Format Options Help
#
# This may also be "None", "All", or any combination of "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", or "MultiViews".
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
#   Options Indexes FollowSymLinks MultiViews
#
# This controls which options the .htaccess files in directories can
# override. Can also be "All", or any combination of "Options", "FileInfo",
# "AuthConfig", and "Limit"
#
#   AllowOverride None
#
# Controls who can get stuff from this server.
#
#   Order allow,deny
#   Allow from all
</Directory>
#
# UserDir: The name of the directory which is appended onto a user's home
# directory if a ~user request is received.
#
<IfModule mod_userdir.c>
#   UserDir public_html
</IfModule>
#
# Control access to UserDir directories. The following is an example
# for a site where these directories are restricted to read-only.
#
#<Directory /home/*/public_html>
#   AllowOverride FileInfo AuthConfig Limit
#   Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
#   <Limit GET POST OPTIONS PROPFIND>
#       Order allow,deny
#       Allow from all
#   </Limit>
#   <LimitExcept GET POST OPTIONS PROPFIND>
#       Order deny,allow
#       Deny from all
#   </LimitExcept>

```

รูปที่ 3.8 module ในส่วนของการตั้งไคเรียกทอริ public\_html

- เมื่อทำการเปลี่ยนที่ UserDir เป็นที่ public\_html แล้วจะแสดงว่าใครเรียกทอริ public\_html ของแต่ละ User เป็นที่เก็บเว็บเพจของตัวเอง ดังรูปที่ 3.9

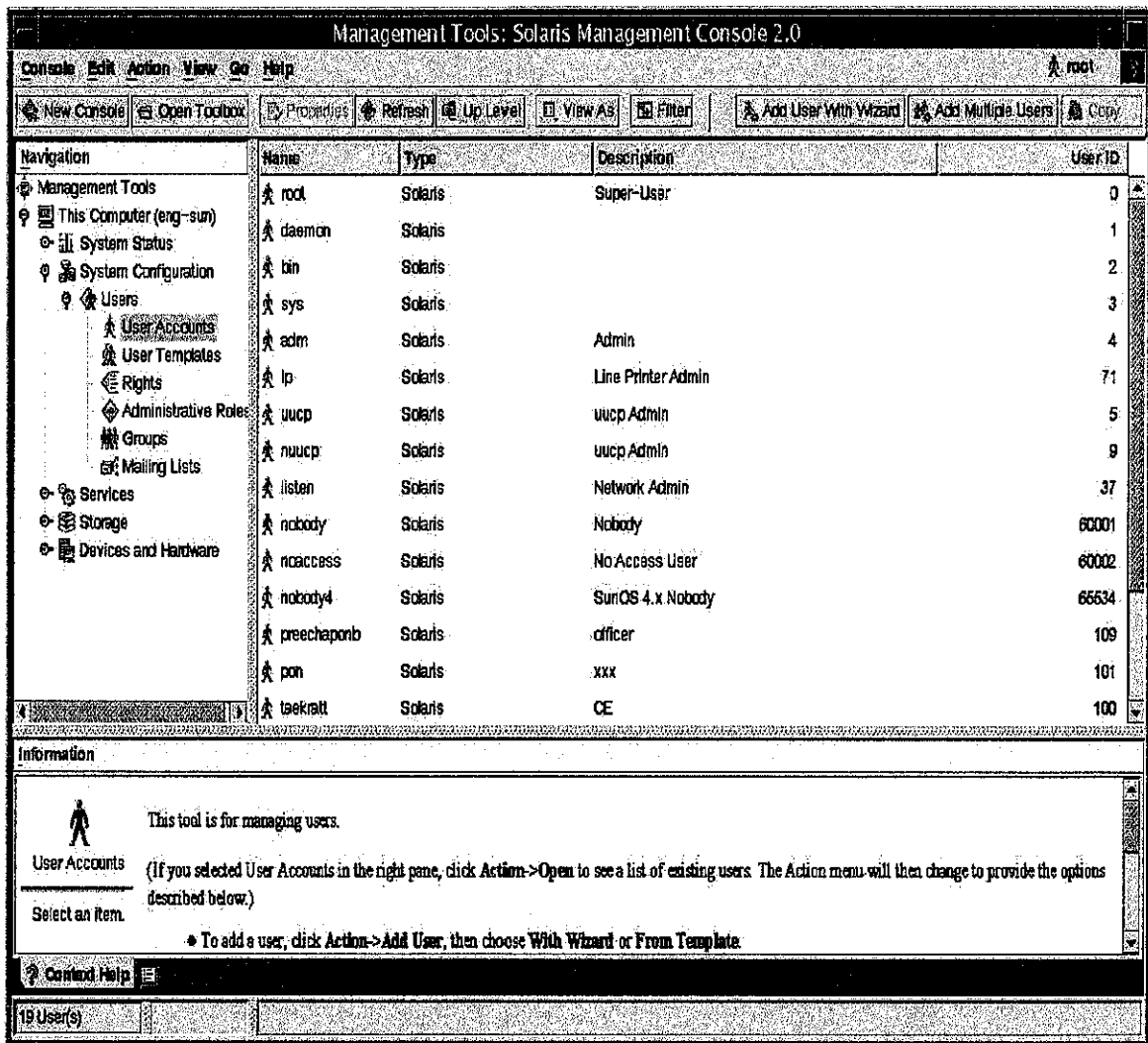


รูปที่ 3.9 ใครเรียกทอริ public\_html ของแต่ละ User ที่เก็บเว็บเพจของตัวเองผ่านทาง Web Browser จากเครื่อง Client หลังจาก ปรับแต่งค่าไฟล์ httpd.conf

**ส่วนของ User กับ Administrator บนระบบปฏิบัติการ Solaris 8**

ในส่วนนี้เป็นส่วนของการจัดการเกี่ยวกับ User โดย Administrators จะ Add User เพื่อเป็นการจองพื้นที่บน Server ของ User แต่ละคน และจัดการสิทธิ์ในการเข้าถึง ของ User แต่ละคน เพื่อให้เป็นการง่ายต่อการบริหารจัดการ โดยมีการแสดงตามขั้นตอนดังนี้

- เมื่อเข้าไปในส่วนของการ Management Tools จะปรากฏหน้าต่าง Solaris Management Console



**รูปนี้ 3.10 หน้าต่าง Management Tools เพื่อให้ ผู้บริหารระบบปรับแต่งระบบ**

- เมื่อคลิกขวาที่ User Accounts จะมีฟังก์ชันในการ Add User ให้ทำการเลือกที่ Add User ก็จะมีฟังก์ชันย่อยสองฟังก์ชันคือ With Wizard และ From Template ในที่นี้จะใช้แบบ Wizard

Management Tools: Solaris Management Console 2.0

Console Edit Action View Go Help

root

New Console Open Toolbox Delete Properties Refresh Up Level View As Filter Add User With Wizard Add Multiple Users

Name	Type	Description	User ID
root	Solaris	Super-User	0
daemon	Solaris		1
bin	Solaris		2
sys	Solaris		3
adm	Solaris	Admin	4
lp			71
uucp		uucp Admin	6
nuuc			9
lstar		Network Admin	37
nobody		Nobody	60001
naccess	Solaris	No Access User	60002
nobody4	Solaris	SunOS 4.x Nobody	65534
preechaponb	Solaris	officer	109
pon	Solaris	xxx	101
taekratt	Solaris	CE	100

Information

This tool is for managing users.

User Accounts (If you selected User Accounts in the right pane, click Action->Open to see a list of existing users. The Action menu will then change to provide the options described below.)

adm

- To add a user, click Action->Add User, then choose With Wizard or From Template.

Context Help

19 User(s)

รูปที่ 3.11 การ Add User

- เมื่อทำการเลือกที่ฟังก์ชัน Add User Wizard จะมีช่องให้ใส่ User Name , Full Name , Description ให้ทำการใส่ User ตามที่ต้องการ แล้วทำการคลิกที่ Next

**Add User Wizard**

**Steps:**

- 1 Enter a user name
- 2 Enter a user identification number.
- 3 Enter the user's password.
- 4 Select the user's primary group.
- 5 Create the user's home directory.
- 6 Specify the mail server.
- 7 Review.

Use this wizard to create a new user account.

Enter a user name and a description for this user.

User Name: tanya

Full Name: tanyapread

Description:

A user name must be unique within a domain; contain 2 to 32 letters and numerals (no spaces or special characters), start with a letter, have at least one lowercase letter.

Back Next Cancel

รูปที่ 3.12 การใส่ Username



- ในส่วนนี้จะแสดงลำดับหมายเลขของ User นั้น โดยอัตโนมัติ ทำการคลิกที่ Next

**Add User Wizard**

**Steps:**

- 1 Enter a user name.
- 2 Enter a user identification number.
- 3 Enter the user's password.
- 4 Select the user's primary group.
- 5 Create the user's home directory.
- 6 Specify the mail server.
- 7 Review.

Enter a unique number to identify this user account.

User ID Number:

The user ID number must be a whole number from 100 to 2147483647.

The number displayed is the next available UID number. Press the arrow buttons to move through the list of numbers, or type in a number yourself.

Back Next Cancel

รูปที่ 3.13 รหัส User

- ในส่วนนี้จะเป็นส่วนที่ให้ทำการใส่รหัสส่วนตัวของแต่ละ User นั้น ทำการคลิกที่ Next

**Add User Wizard**

**Steps:**

- 1 Enter a user name.
- 2 Enter a user identification number.
- 3 Enter the user's password.
- 4 Select the user's primary group.
- 5 Create the user's home directory.
- 6 Specify the mail server.
- 7 Review.

You may choose to let the user set the password, or set it yourself now.

User Must Set Password At First Login

User Must Use This Password At First Login:

Password:

Confirm Password:

The password must contain a combination of 6 to 15 characters (although only the first 8 are used). Within the first 6, there must be at least 2 alphabetic characters and at least one number. The password is case sensitive.

Note: If you are running in an NIS+ domain, you must provide a password.

**Back** **Next** **Cancel**

รูปที่ 3.14 การใส่ Password

- ในส่วนนี้จะเป็นส่วนที่ให้ทำการเลือก Group ของแต่ละ User โดยแต่ละ Group จะมีสิทธิในการเข้าถึงระบบที่แตกต่างกัน

**Add User Wizard**

**Steps:**

- 1 Enter a user name.
- 2 Enter a user identification number.
- 3 Enter the user's password.
- 4 **Select the user's primary group.**
- 5 Create the user's home directory.
- 6 Specify the mail server.
- 7 Review.

Choose a group from the drop-down list:

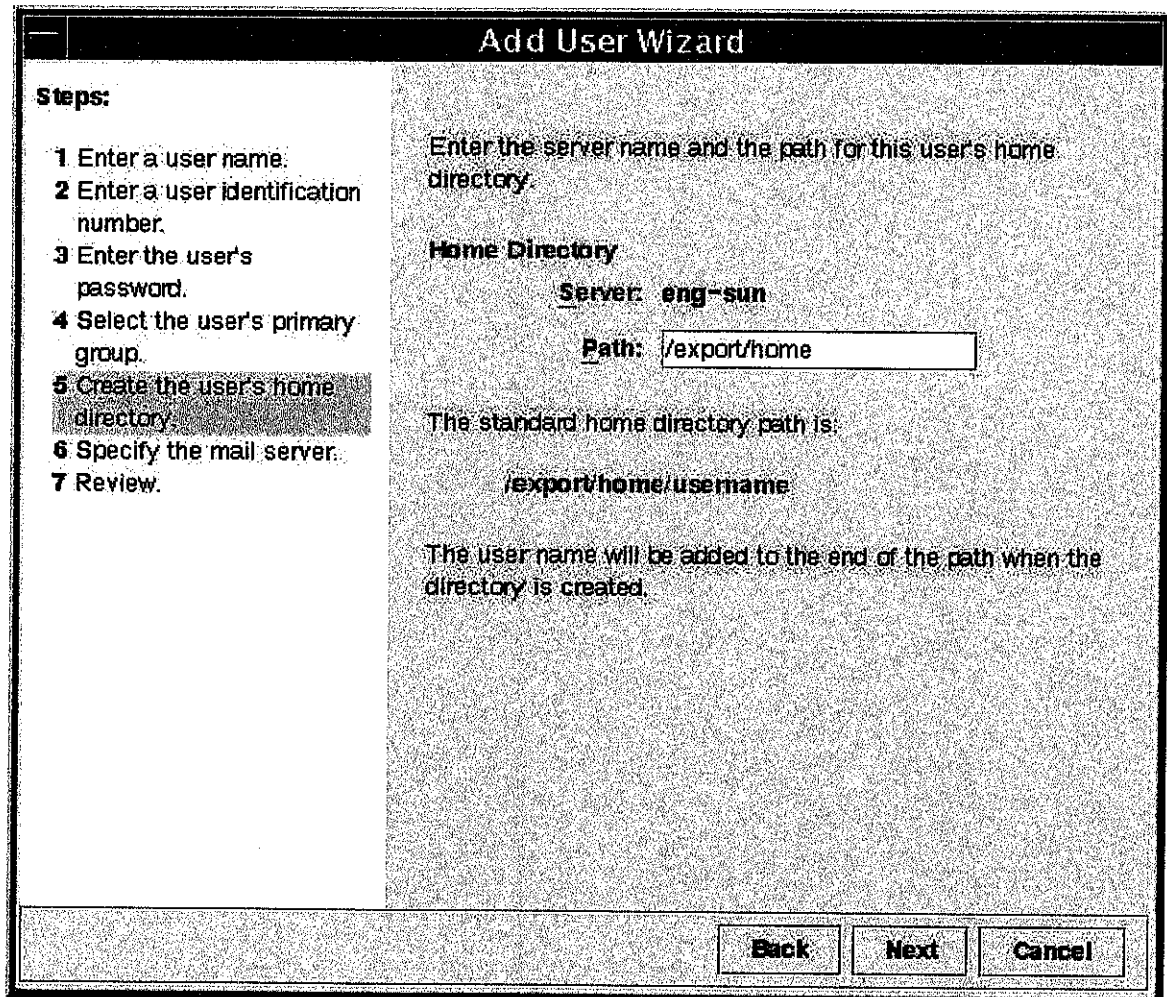
**Primary Group:**

The operating system associates the primary group with all files created by the user. This permits file-sharing with other members of the group.

**Back** **Next** **Cancel**

รูปที่ 3.15 การเลือก Group

- ในส่วนนี้จะเป็นส่วนที่แสดง Home ไคเร็กทอรีของแต่ละ User ซึ่งจะถูกเก็บอยู่ที่ path:  
/export/home



รูปที่ 3.16 Home ไคเร็กทอรีของแต่ละ User ซึ่งจะถูกเก็บอยู่ที่ path:

/export/home

- เมื่อทำการใส่ข้อมูลทั้งหมดเรียบร้อยแล้ว จะมีการแสดงข้อมูลทั้งหมดนั้นเพื่อตรวจสอบความถูกต้อง จากนั้นให้ทำการคลิก Finish เป็นการสิ้นสุดวิธีการ Add User

**Add User Wizard**

**Steps:**

- 1 Enter a user name.
- 2 Enter a user identification number.
- 3 Enter the user's password.
- 4 Select the user's primary group.
- 5 Create the user's home directory.
- 6 Specify the mail server.
- 7 **Review**

This is the information you provided for the new user account.

User Name:

Full Name:

Description:

User ID Number:

Primary Group:

Home Directory

Server:

Path:

Mail Server:

To create the new user account, click Finish.  
To change an entry, click back to it.  
Later, change entries with the User Properties dialog.

รูปที่ 3.17 รายละเอียดที่กรอกทั้งหมด