

บทที่ 9

การกำหนดให้เครื่องลูกข่ายทำงานแบบ Thin client

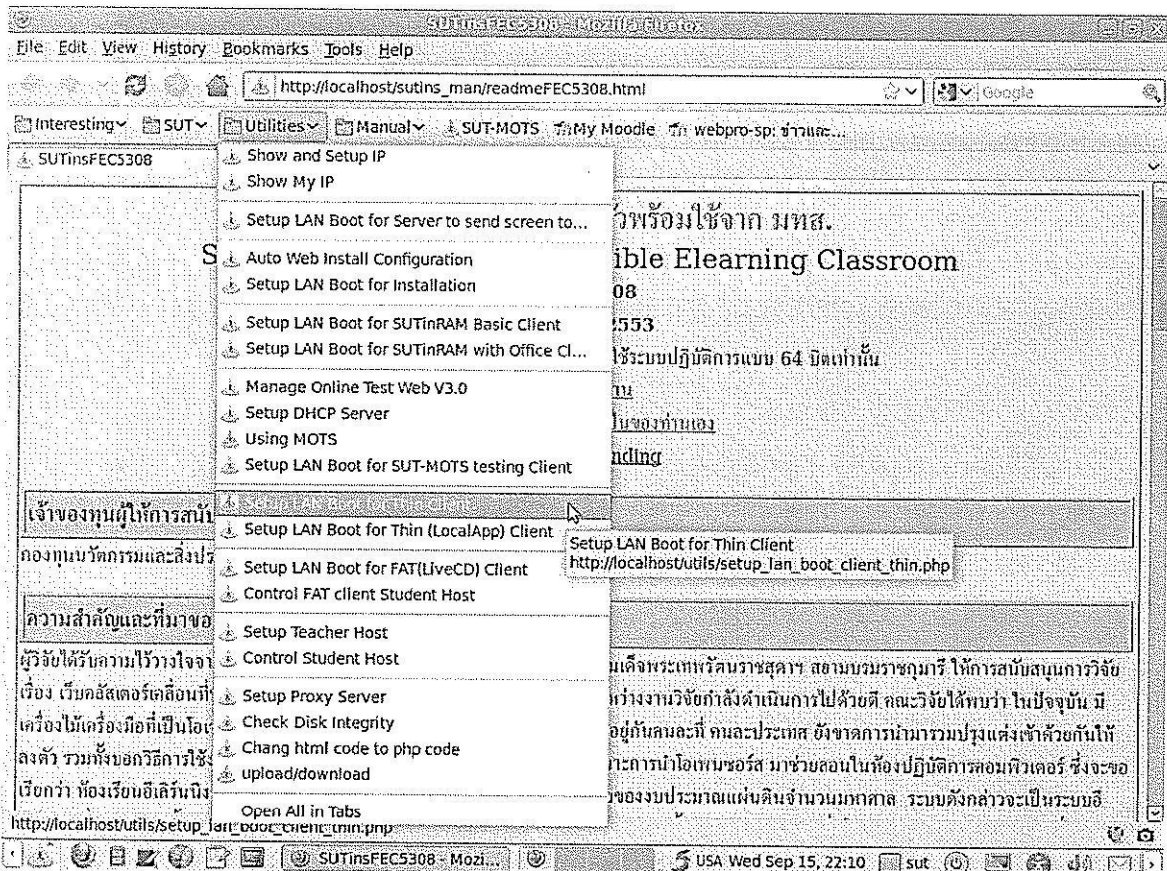
ขั้นตอนการทำงานมีดังนี้

9.1 กำหนดให้เครื่องเซิร์ฟเวอร์มีไอพีเป็น 172.16.0.1

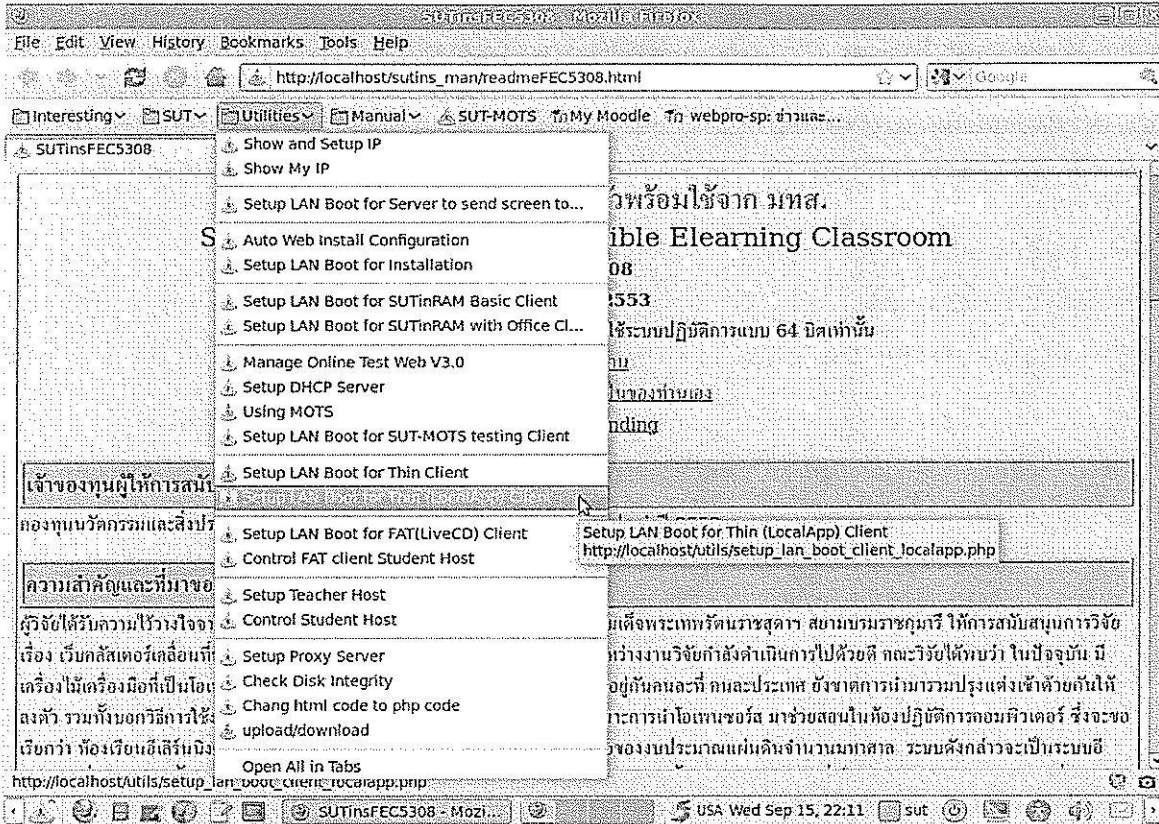
ดังแสดงไว้แล้วในบทที่ 5 ตอนต้น

9.2 กำหนดให้เครื่องลูกข่ายทำงานแบบ Thin client

เริ่มด้วยการคลิกที่เมนูของ Firefox ที่ Utilities-->Setup LAN Boot for Thin Client ดังรูปที่ 9.1 หรือ Utilities-->Setup LAN Boot for Thin (LocalApp) Client ดังรูปที่ 9.2 ซึ่งมีการทำงานคล้ายกัน ต่างกันเพียงเล็กน้อย นั่นคือเมื่อเครื่องลูกข่ายทำงานเป็น Thin client ทุกอย่างจะรันที่เครื่องเซิร์ฟเวอร์ แต่ถ้าเครื่องลูกข่ายทำงานเป็น Thin client(Local App) ทุกอย่างจะรันที่เครื่องเซิร์ฟเวอร์ยกเว้น 4 โปรแกรมต่อไปนี้จะรันที่เครื่องลูกข่าย นั่นคือโปรแกรม Firefox, Gimp(สำหรับตกแต่งภาพ), OpenOffice และ VLC(สำหรับเล่นไฟล์วีดีโอ)

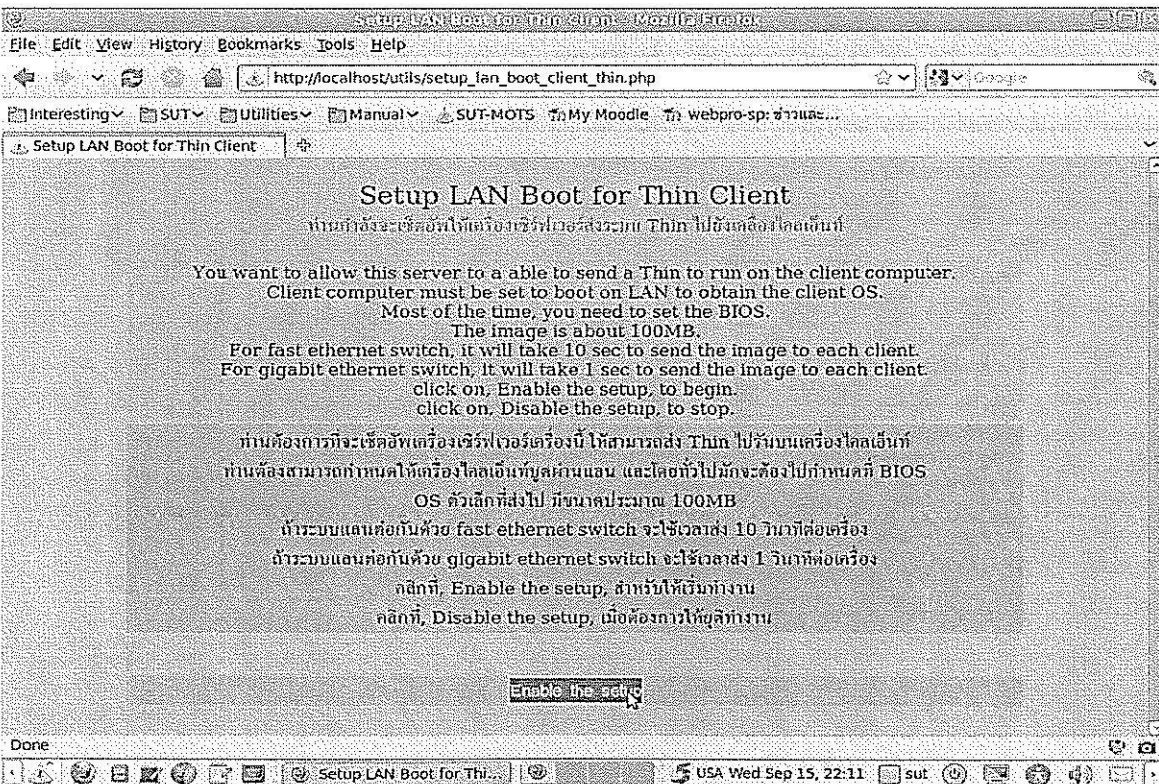


รูปที่ 9.1 เมนูกำหนดให้เครื่องลูกข่ายทำงานแบบ Thin client



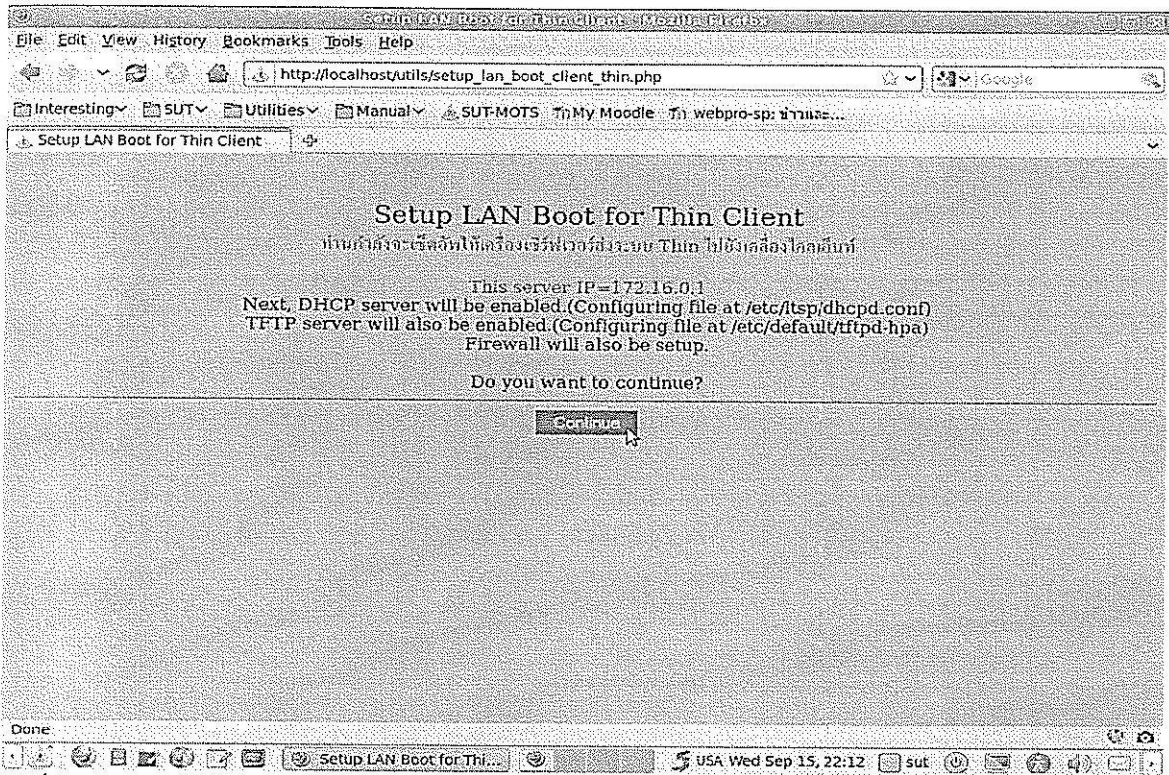
รูปที่ 9.2 เมนูกำหนดให้เครื่องลูกข่ายทำงานแบบ Thin client ที่มีบางโปรแกรม ไปรันบนเครื่องลูกข่าย

สมมุติเราเลือกเมนูแรก ดังรูปที่ 9.1 เราก็จะเข้าสู่รายละเอียดการเซตอัพดังรูปที่ 9.3 ซึ่งเมื่ออ่านรายละเอียดจนเข้าใจแล้ว ให้คลิก Enable the setup



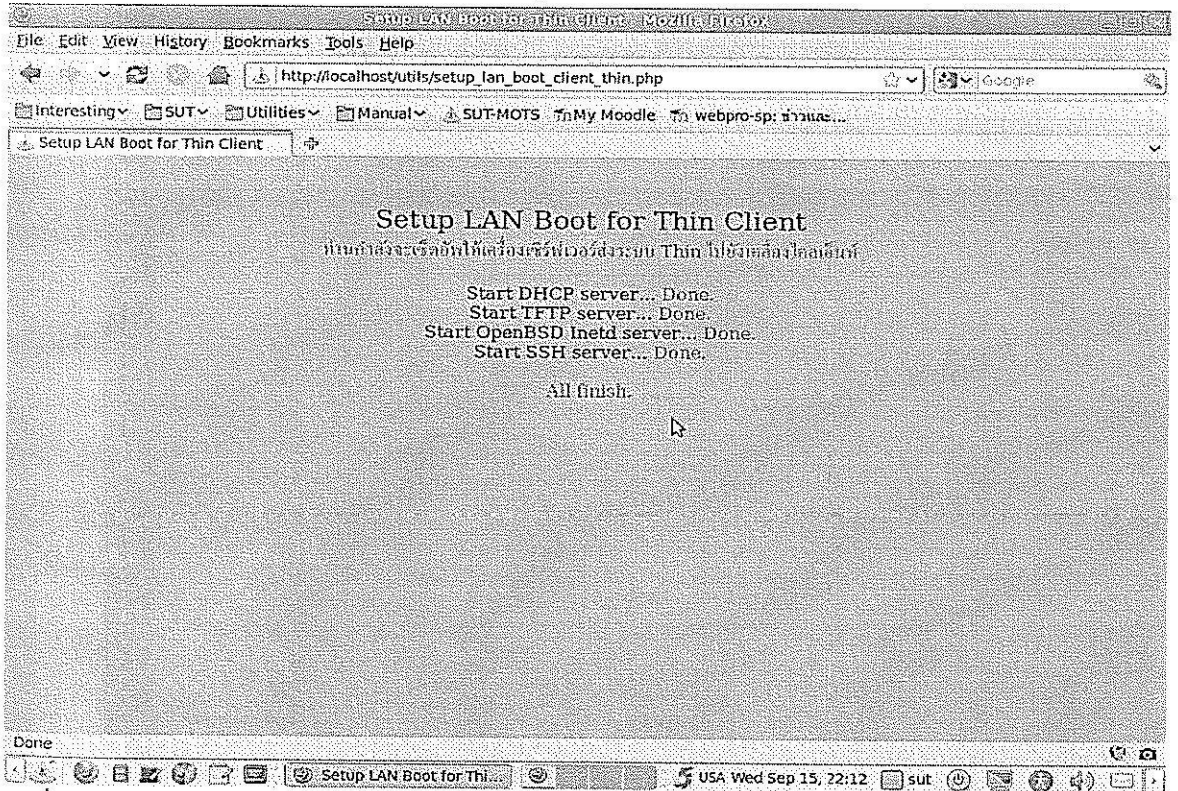
รูปที่ 9.3 รายละเอียดการทำงานของโหมด Thin Client

จากนั้นจะเข้าสู่หน้าจอดังรูปที่ 9.4 ซึ่งจะมีการตรวจสอบไอพีของเครื่องเซิร์ฟเวอร์ อันเป็น 172.16.0.1 และเมื่อผู้ใช้แน่ใจ จะทำงานต่อ ให้คลิกที่ Continue



รูปที่ 9.4 ตรวจสอบ ไอพีและให้ยืนยันที่จะทำงานต่อ

จากนั้น โปรแกรมจะทำการเซตอัพต่าง จนเสร็จสิ้นดังรูปที่ 9.5



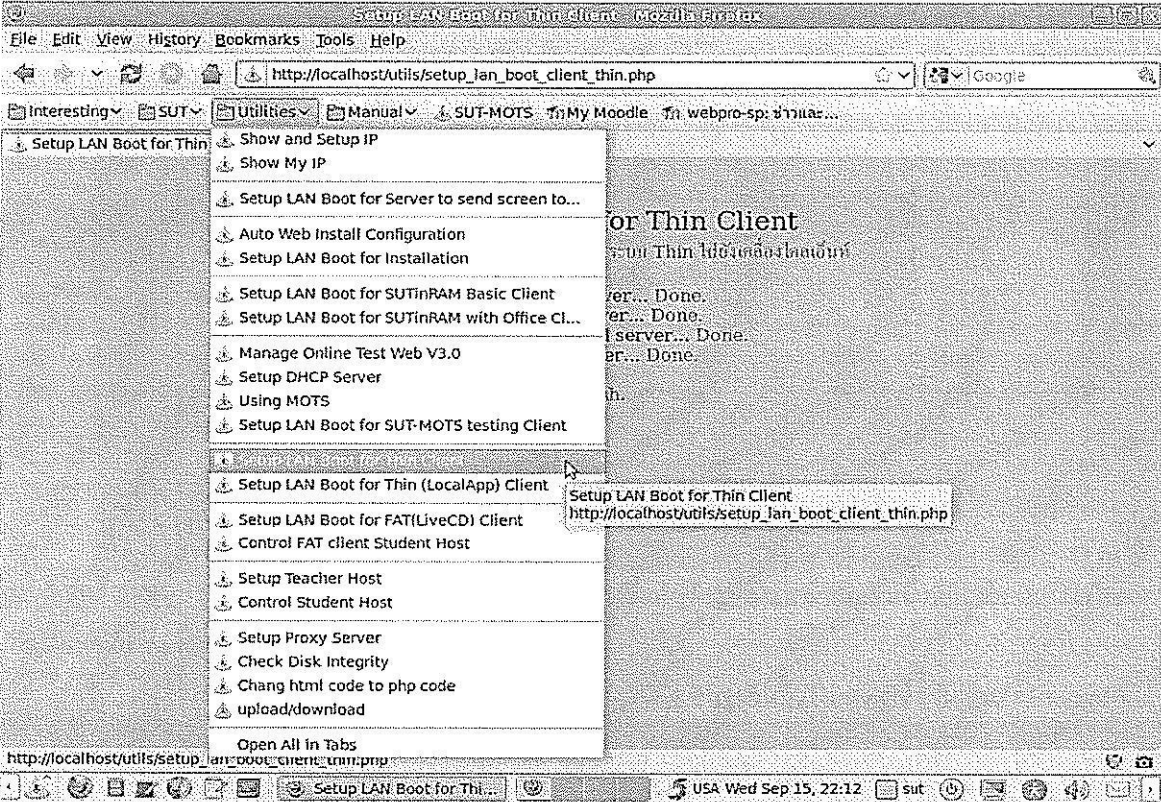
รูปที่ 9.5 ดำเนินการกำหนดให้เครื่องลูกข่ายรันแบบ Thin Client

9.3 ดำเนินการบูตเครื่องลูกข่ายผ่านแลน

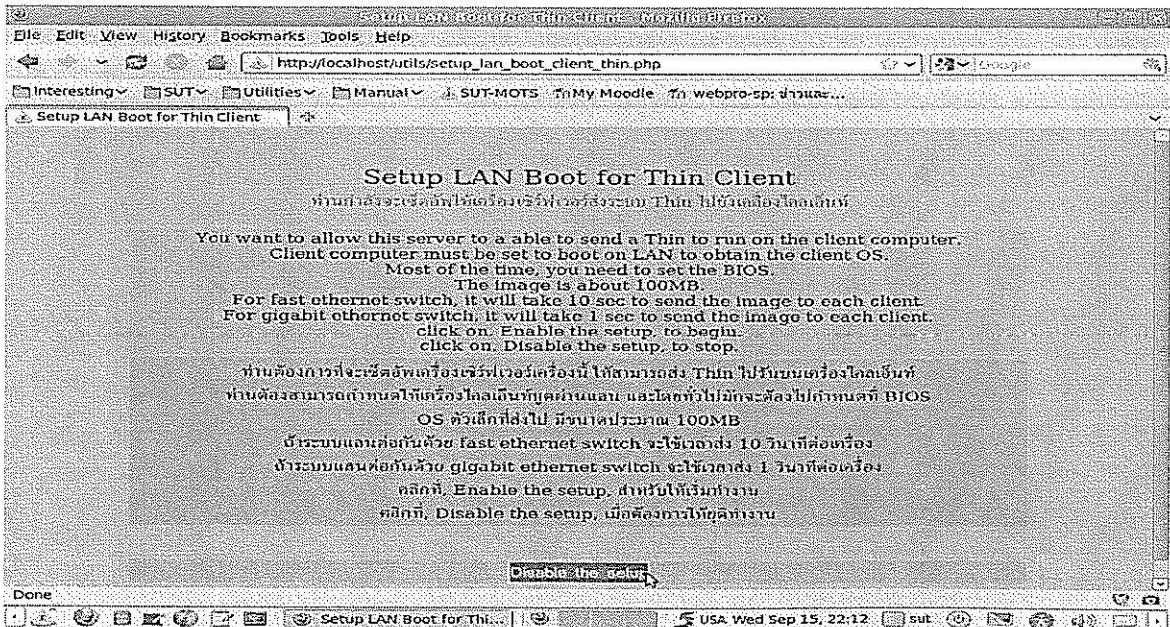
ให้บูตเครื่องลูกข่ายทั้งหมดที่ต้องการดำเนินงานตามที่ตั้งไว้ ผ่านแลน เครื่องลูกข่ายก็จะทำตัวเป็น Thin Client คือรันทุกโปรแกรมที่เครื่องเซิร์ฟเวอร์ ดังนั้นการที่จะทำงานในโหมดนี้ จำเป็นที่เครื่องเซิร์ฟเวอร์จะต้องมีประสิทธิภาพสูง จึงจะรองรับการทำงานของเครื่องลูกข่ายได้จำนวนมาก

9.4 การยุติการทำงานของโหมดนี้

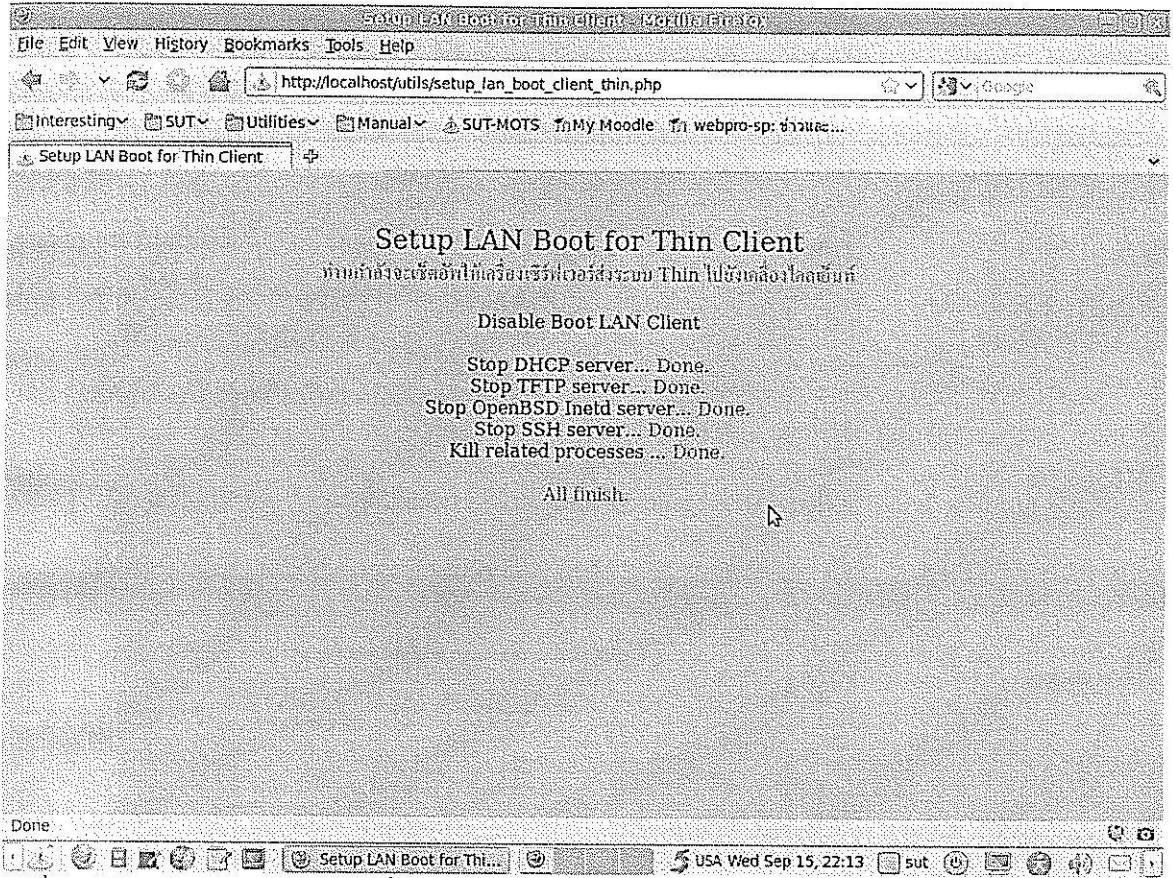
หลังจากการใช้งานเสร็จสิ้นแล้ว ควรยุติการทำงานของโหมดนี้ โดยดำเนินการตามรูปที่ 9.6, 9.7 และ 9.8 ตามลำดับ



รูปที่ 9.6 เมื่อกำหนดให้เครื่องลูกข่ายทำงานแบบ Thin client



รูปที่ 9.7 หน้าต่างแสดงปุ่มให้หยุดการทำงาน



รูปที่ 9.8 ชุดการทำงานของโหมคให้เครื่องลูกข่ายรันแบบ Thin Client

บทที่ 10

การกำหนดให้เครื่องลูกข่ายทำงานแบบ FAT client

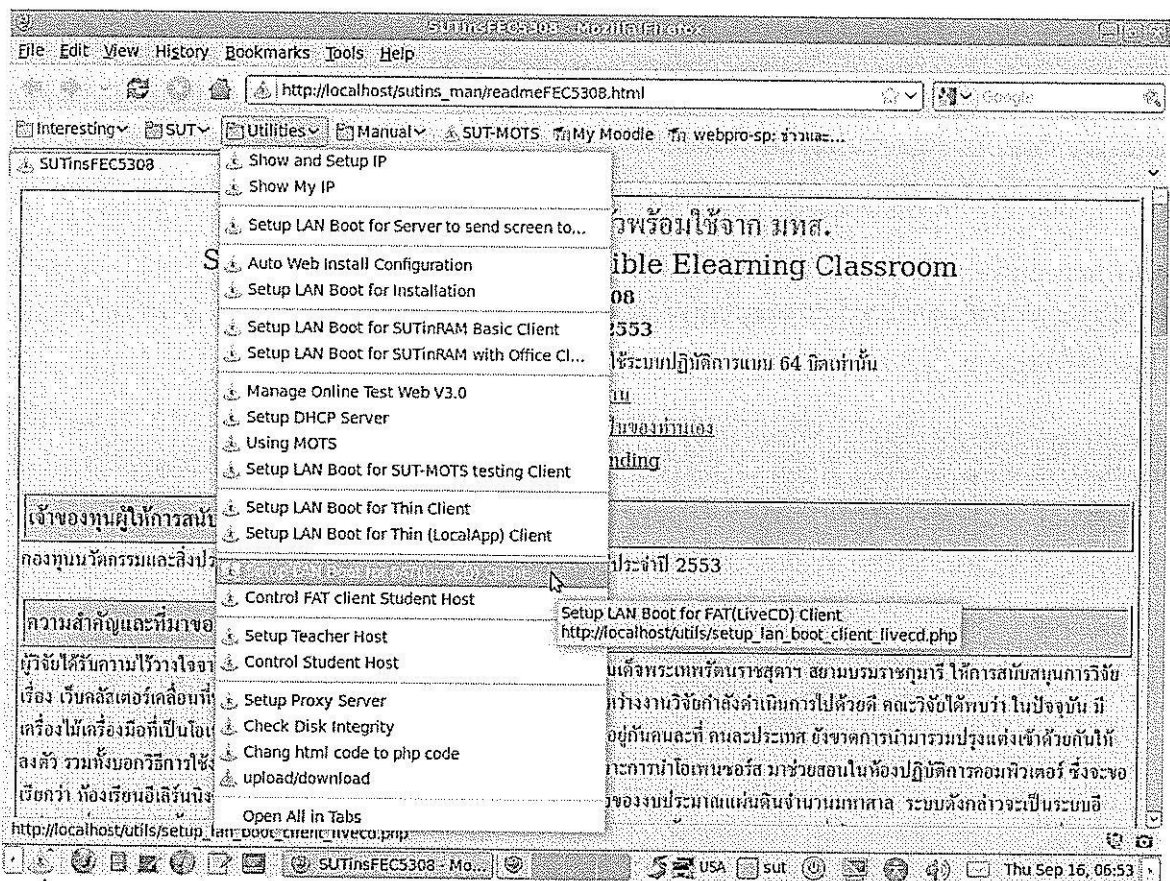
ขั้นตอนการทำงานมีดังนี้

10.1 กำหนดให้เครื่องเซิร์ฟเวอร์มีไอพีเป็น 172.16.0.1

ดังแสดงไว้แล้วในบทที่ 5 ตอนต้น

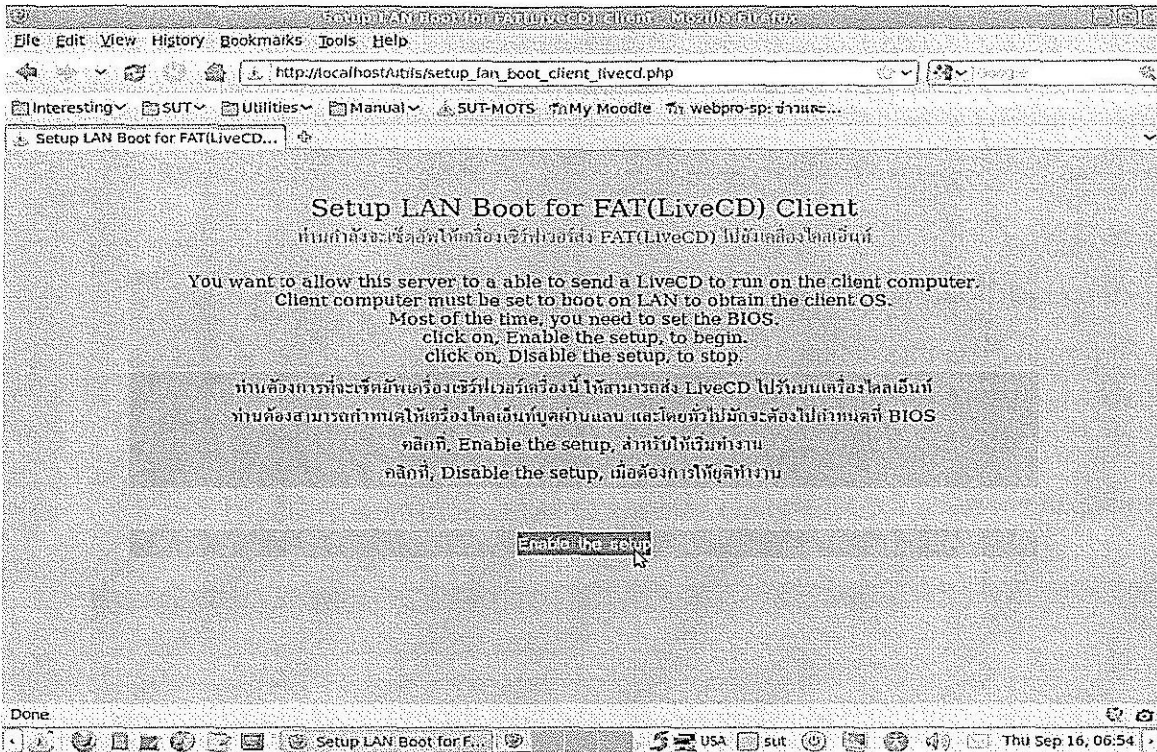
10.2 กำหนดให้เครื่องลูกข่ายทำงานแบบ FAT client

เริ่มด้วยการคลิกที่เมนูของ Firefox ที่ Utilities-->Setup LAN Boot for FAT(LiveCD) Client ดังรูปที่ 10.1



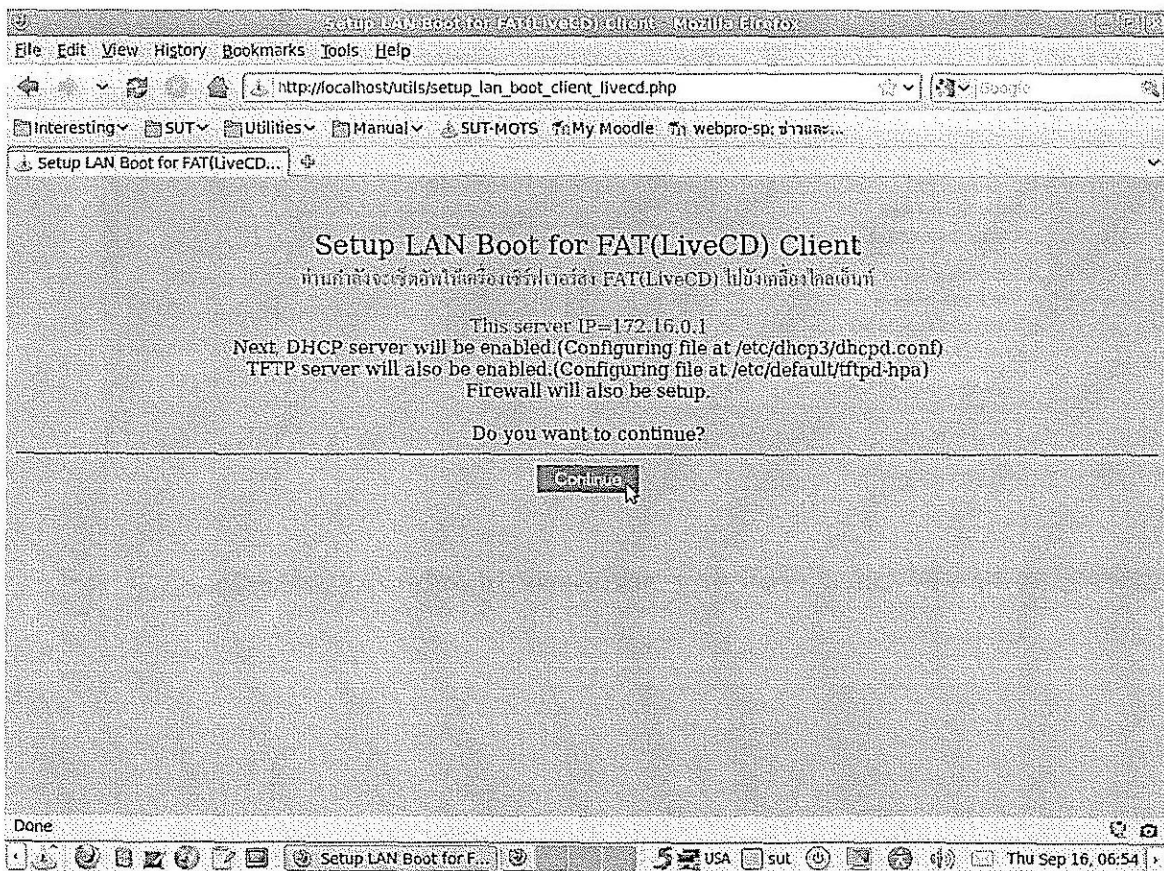
รูปที่ 10.1 เมนูเลือกโหมดให้เครื่องลูกข่ายทำงานแบบ FAT client

เราก็จะเข้าสู่รายละเอียดการเซ็ตอัพดังรูปที่ 10.2 ซึ่งเมื่ออ่านรายละเอียดจนเข้าใจแล้ว ให้คลิก Enable the setup



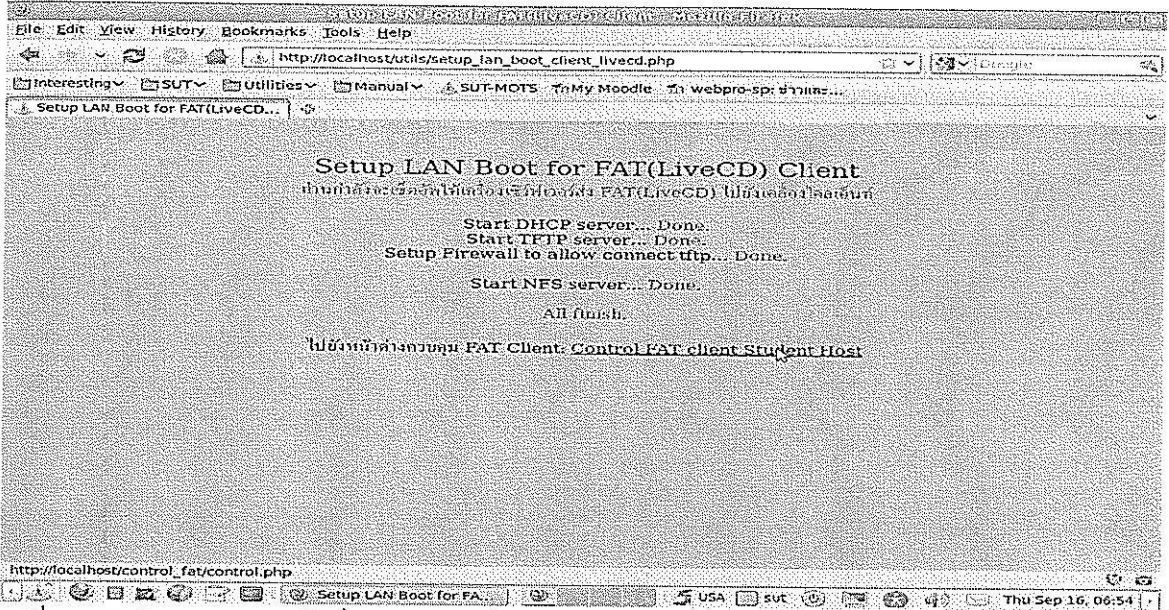
รูปที่ 10.2 กำหนดให้เครื่องลูกข่ายเป็นแบบ FAT client

จากนั้นจะเข้าสู่หน้าจอจากรูปที่ 10.3 ซึ่งจะมีการตรวจสอบไอพีของเครื่องเซิร์ฟเวอร์ ด้านเป็น 172.16.0.1 และผู้ใช้งานใจ ก็ให้ทำงานต่อโดยคลิกที่ Continue



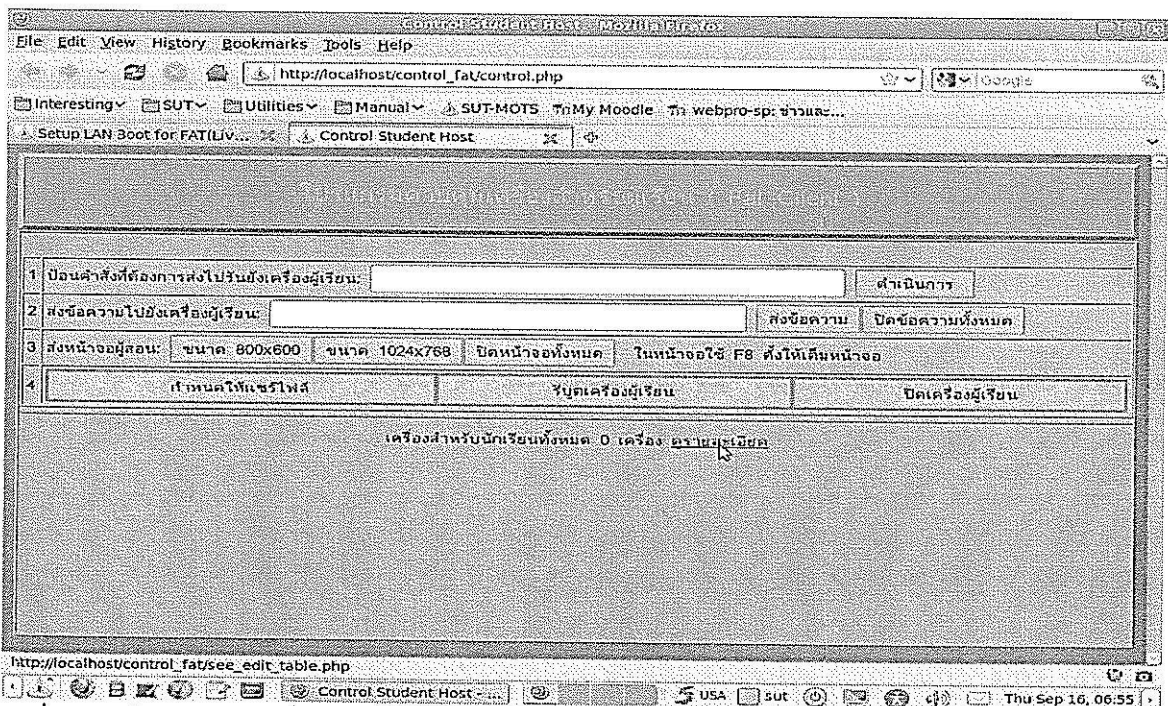
รูปที่ 10.3 ตรวจสอบไอพีและให้ยืนยันที่จะทำงานต่อ

จากนั้น โปรแกรมจะทำการเชื่อมต่อต่าง จนเสร็จสิ้นดังรูปที่ 10.4



รูปที่ 10.4 คำเนิการกำหนดให้เครื่องลูกข่ายรันเป็น Thin Client

ถ้าต้องการตั้งงาน ไปยังเครื่องลูกข่ายที่เป็น FAT Client ด้วย ก็ให้คลิกที่ลิงก์ Control FAT client Student Host ที่ปรากฏในรูปที่ 10.4 จากนั้นจะเข้าสู่หน้าต่างควบคุม ดังรูปที่ 10.5 ในหน้าต่างนี้ สามารถสั่งงานได้หลายรูปแบบ ดังใน ส่วนแรก เป็นการตั้งค่าตั้ง ไปรันที่เครื่องลูกข่าย ส่วนที่ 2 เป็นการส่งข้อความ ไปแสดงที่หน้าจอของวเครื่องลูกข่าย ส่วนที่ 3 เป็นการส่งหน้าจอผู้สอน ไปแสดงที่เครื่องลูกข่ายด้วยขนาดต่างๆ โดย บนหน้าจอที่ส่งไปนั้นสามารถใช้ F8 ตั้งให้แสดงเต็มจอหรือย่อส่วนลงก็ได้ ส่วนที่ 4 เป็นการเปิดให้มีการแชร์ข้อมูลระหว่างเครื่องเซิร์ฟเวอร์และเครื่องลูกข่ายทั้งหมด รวมทั้งระหว่างเครื่องลูกข่ายด้วยกัน อีกสองปุ่มเพื่อสั่งรีบูตและเพื่อสั่งปิดเครื่องลูกข่ายทั้งหมดพร้อมกัน ด้านล่างสุดของจอ จะแสดงตัวเลขของจำนวนเครื่องลูกข่ายที่กำลังทำงาน ในโหมด FAT client อยู่ และสามารถตรวจดูไอพีของเครื่องเหล่านี้ได้จากการคลิกที่ลิงก์ ดูรายละเอียด



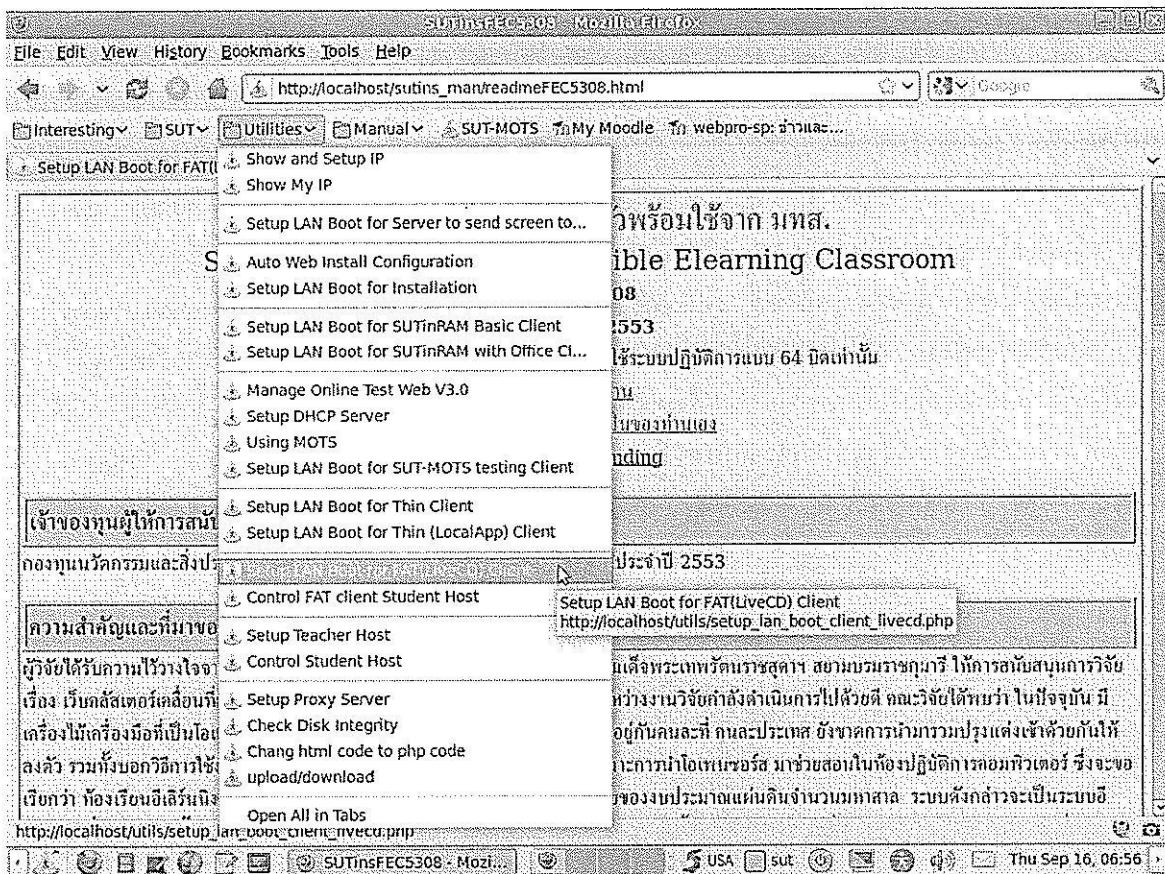
รูปที่ 10.5 หน้าต่างควบคุมเครื่องลูกข่ายที่เป็น FAT Client

10.3 คำเนิการบูตเครื่องถูกข่ายผ่านแลน

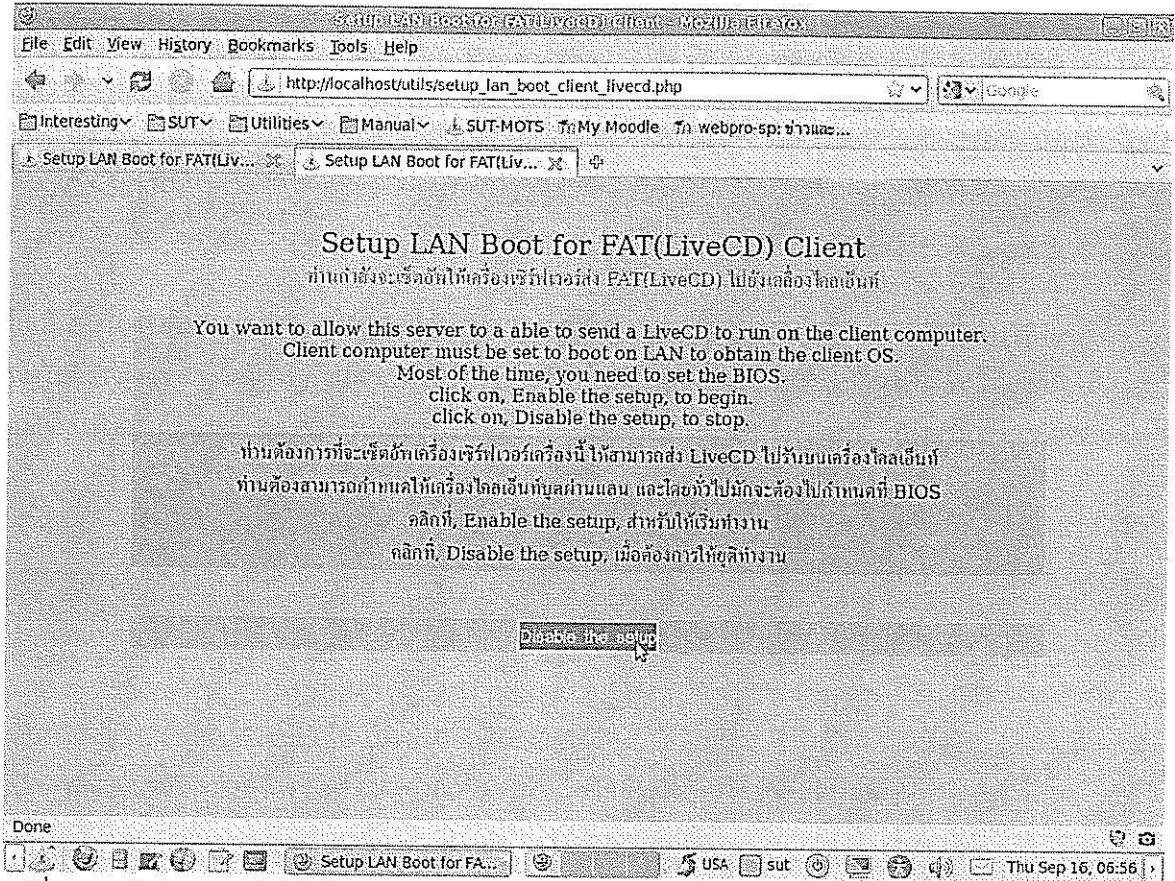
ให้บูตเครื่องถูกข่ายทั้งหมดที่ต้องการดำเนินงานตามที่ตั้งไว้ ผ่านแลน เครื่องถูกข่ายก็จะทำตัวเป็น FAT Client คือทั้งระบบปฏิบัติการและโปรแกรมต่างๆ จะถูกโหลดมาจากเครื่องเซิร์ฟเวอร์ แล้วทำงานบนเครื่องถูกข่ายแต่ละเครื่องอย่างอิสระ ดังนั้นการใช้งานโหมคนี่ เครื่องเซิร์ฟเวอร์ไม่จำเป็นต้องมีประสิทธิภาพสูงมากนัก แต่ต้องการความเร็วในการเชื่อมต่อผ่านระบบแลนสูง นั่นคือควรใช้เป็นจิกบิตแลน จึงจะรองรับการทำงานของเครื่องถูกข่ายได้จำนวนมาก

10.4 การยุติการทำงานของโหมคนี่

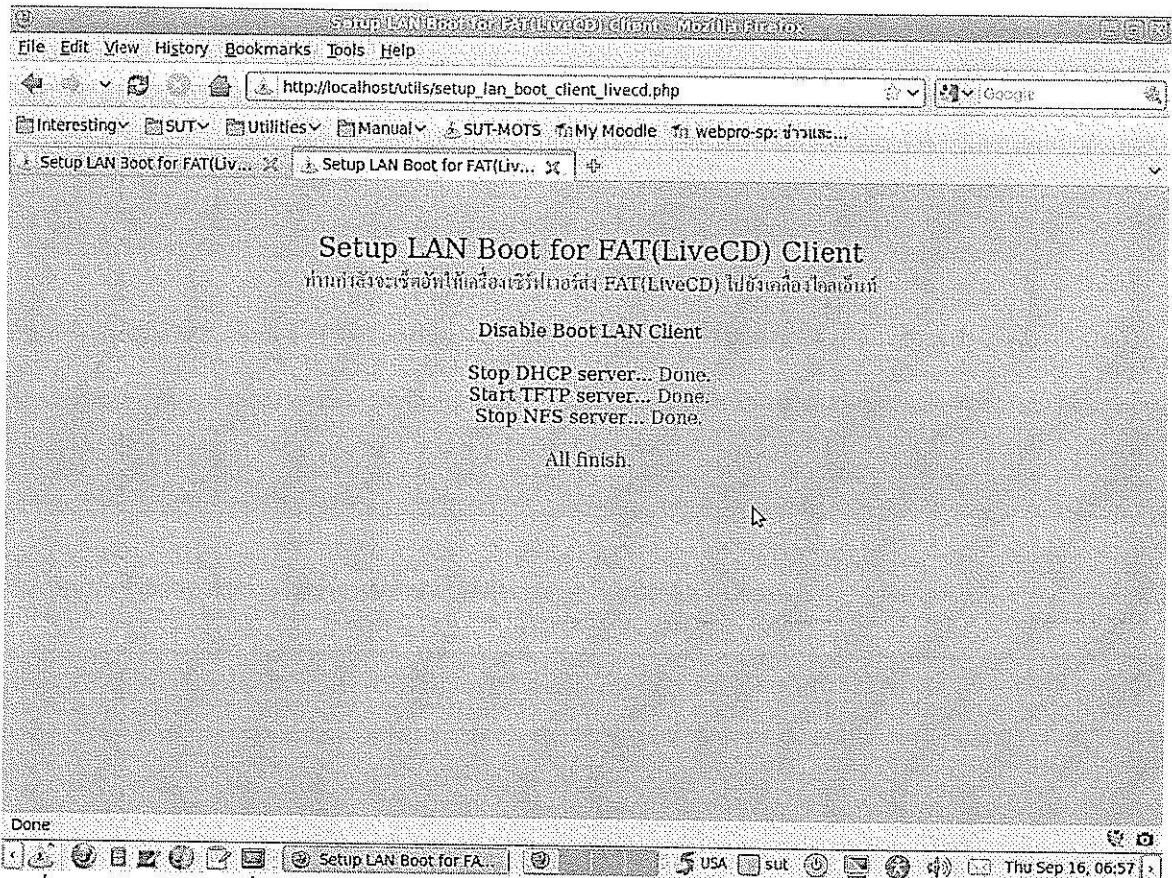
หลังจากการใช้งานเสร็จแล้ว ควรยุติการทำงานของโหมคนี่ โดยดำเนินการตามรูปที่ 10.6, 10.7 และ 10.8 ตามลำดับ



รูปที่ 10.6 โหมคการส่งระบบปฏิบัติการตัวเล็กไปรันในหน่วยความจำที่เครื่องถูกข่าย



รูปที่ 10.7 หน้าต่างแสดงปุ่มให้เลือกการทำงาน



รูปที่ 10.8 ยุติโหมดให้เครื่องลูกข่ายรันแบบ FAT Client

บทที่ 11

การปรับแต่ง FAT Client, Thin Client และ Thin (Local app) Client

1. การปรับแต่ง FAT Client

ติดตั้ง SUTinsDesktop5306 ในพาร์ทิชันที่ต้องการ (สมมุติว่าติดตั้งอยู่ที่ sda15)

เข้าไปปรับแต่งตามที่ต้องการเช่น ถอนบางโปรแกรมหรือเพิ่มเติมบางโปรแกรมและเปลี่ยนพื้นเคสท์ท็อป เป็นต้น

และต้องทำการติดตั้ง package เหล่านี้เพิ่มเข้าไปใน SUTinsDesktop5306 ด้วยจึงจะสามารถนำมาทำเป็น FAT Client ได้

```
sudo apt-get install --yes ubuntu-standard casper lupin-casper
sudo apt-get install --yes discoverl laptop-detect os-prober
sudo apt-get install --yes linux-generic
```

เมื่อติดตั้งเสร็จแล้วให้บูตระบบเข้ามาใน SUTinsFEC5308

เปิด Terminal ขึ้นมาการสร้าง FAT Client นั้นต้องทำในสิทธิ์ของ root ให้ใช้คำสั่งต่อไปนี้

```
sut@sut:~$ sudo su
```

ทำการเปลี่ยนไดเรกทอรีที่ทำงานไปยัง /var/www/utills พร้อมทั้งสร้างไดเรกทอรีชื่อ fat และทำการ mount ไปยัง sda15

```
root@sut:/home/sut# cd /var/www/utills/
root@sut:/var/www/utills# mkdir fat
root@sut:/var/www/utills# mount /dev/sda15 fat/
```

คัดลอก vmlinuz และ initrd จาก SUTinsDesktop 5306 ที่เราได้ทำการปรับแต่งแล้ว

```
root@sut:/var/www/utills# cp fat/boot/vmlinuz-2.6.**-**-generic lan_boot_client_lived/casper/vmlinuz
root@sut:/var/www/utills# cp fat/boot/initrd.img-2.6.**-**-generic lan_boot_client_lived/casper/initrd.gz
```

ขั้นตอนสุดท้ายคือการสร้างไฟล์ระบบที่จะไว้ใช้สำหรับบูตในแบบ FAT Client

```
root@sut:/var/www/utills# rm -f lan_boot_client_lived/casper/filesystem.squashfs
root@sut:/var/www/utills# mksquashfs fat lan_boot_client_lived/casper/filesystem.squashfs -e boot
root@sut:/var/www/utills# printf $(sudo du -sx --block-size=1 fat | cut -f1) > lan_boot_client_lived/casper/filesystem.size
root@sut:/var/www/utills# cd lan_boot_client_lived; find . -type f -print0 | xargs -0 md5sum | grep -v "\/md5sum.txt" > md5sum.txt
root@sut:/var/www/utills/lan_boot_client_lived# chown www-data.www-data -R *; chmod 755 -R *; exit
```

เสร็จขั้นตอนนี้ทั้งหมดก็สามารถให้เครื่อง Client บุกผ่าน Network เพื่อใช้งาน FAT Client ตัวที่เราปรับแต่งใหม่แล้ว

2. การปรับแต่ง Thin Client

สำหรับ SUTinsFEC 5308 นั้น ได้เตรียมระบบเซิร์ฟเวอร์ของ Thin Client ไว้เรียบร้อยแล้ว

```
sudo apt-get install ltsp-server-standalone openssh-server
```

ในส่วนของการสร้าง image สำหรับ client ใช้คำสั่งดังต่อไปนี้

```
sudo ltsp-build-client -a i386
```

การทำให้เครื่อง thin client สามารถ Login เข้าระบบได้โดยอัตโนมัติ ต้องทำตามขั้นตอนดังนี้ เราต้องเพิ่ม user เข้าไปในระบบก่อนในตัว SUTinsFEC ได้ทำการเพิ่มเอาไว้แล้ว

แก้ไขหรือสร้างไฟล์ `/opt/ltsp-localapp/i386/etc/ltsc.conf` ให้มีข้อมูลดังนี้

```
[Default]
LDM_ALLOW_GUEST = True

[172.16.1.1]
LDM_USERNAME = client001
LDM_PASSWORD = 123456
```

หลังจากนั้นทำการ update ตัว image ของ thin client

```
sudo ltsp-update-image -a i386 -b /opt/ltsp
```

3. การปรับแต่ง Thin Client (LocalApp)

การเตรียมระบบ thin client แบบ LocalApp จะคล้าย ๆ กับของ thin client แต่เปลี่ยนที่เก็บใหม่

```
sudo ltsp-build-client -a i386 -b /opt/ltsp-localapp
```


ติดตั้งโปรแกรมที่ต้องการให้ไปรันในเครื่องลูกตามที่ต้องการ

```
sudo chroot /opt/ltsp-localapp/i386 apt-get install firefox flashplugin-installer gimp
sudo chroot /opt/ltsp-localapp/i386 apt-get update
```

จะเหมือนกับการทำ thin client ธรรมดาต่างกันอยู่ที่ว่าเราจะติดตั้งโปรแกรมที่เราต้องการให้รันในเครื่องของ client เพิ่มเข้าไปเช่น

เราต้องการให้เครื่อง client รัน Firefox, GIMP เป็นต้น

```
[Default]

LOCAL_APPS=True
LOCAL_APPS_MENU = True
LOCAL_APPS_MENU_ITEMS = firefox, gimp

LDM_ALLOW_GUEST = True

[[172.16.1.1]
LDM_USERNAME = client001
LDM_PASSWORD = 123456
```

เสร็จแล้วทำการ update image ของตัว thin client แบบ localapp

```
sudo ltsp-update-image -a i386 -b /opt/ltsp-localapp
```

บทที่ 12

NIC Bonding

การทำ NIC bonding หรือ port bonding คือการกำหนดให้ NIC หรือการ์ดแลน 2 ตัว(ขึ้นไป)ทำงานร่วมกันด้วยหมายเลขไอพีเดียวกันและให้มีการสลับกันรับส่งข้อมูล เมื่อทำเช่นนี้จะทำให้การรับส่งข้อมูลผ่านทางการ์ดแลนทำได้เร็วขึ้น ตัวอย่างเช่น เมื่อยังไม่ทำ NIC bonding แต่ละการ์ดแลนอาจจะส่งหรือรับข้อมูลด้วยความเร็ว 100 เม็กกะบิตในหนึ่งวินาที ที่หมายเลขไอพีหนึ่งของเครื่องเซิร์ฟเวอร์ แต่หลังจากนำการ์ดแลนสองตัว มาทำ NIC bonding แล้ว ความเร็วรวมในการรับส่งข้อมูลจะกลายเป็น 200 เม็กกะบิตในหนึ่งวินาทีเป็นต้น ความสามารถเช่นนี้จะทำให้ เครื่องเซิร์ฟเวอร์ของเราสามารถรับส่งข้อมูลผ่านทางการ์ดแลน ได้เร็วขึ้น อันจะเป็นผลทำให้ โหมด Thin client, Thin client แบบ Local app, FAT client, send screen, auto install, web install ทำงานได้เร็วขึ้นเป็นเท่า(หรือหลายเท่า)ตัว ทั้งนี้ขึ้นอยู่กับจำนวน port ที่นำมา bond เข้าด้วยกัน

อ่านคู่มือเพิ่มเติมได้จาก

<https://help.ubuntu.com/community/UbuntuLTSP/Trunking>

<http://linuxshellaccount.blogspot.com/2009/01/setting-up-dual-dual-nic-bonding-on.html>

ขั้นตอนการทำให้ดังนี้

1. ติดตั้ง package เพิ่มเติม

```
sudo apt-get install ifenslave
```

ท่านไม่ต้องทำเพราะติดตั้งไว้ให้แล้ว

2. แก้ไขไฟล์ /etc/modprobe.d/aliases ด้วยคำสั่ง sudo gedit /etc/modprobe.d/aliases

```
alias bond0 bonding
```

```
options bonding mode=0 miimon=100
```

3. แก้ไขไฟล์ /etc/network/interfaces ด้วยคำสั่ง sudo gedit /etc/network/interfaces

```
auto lo
```

```
iface lo inet loopback
```

```
auto bond0
```

```
iface bond0 inet static
```

```
address 192.168.1.111
```

```
netmask 255.255.255.0
```

```
network 192.168.1.0
```

```
gateway 192.168.1.1
```

```
slaves eth0 eth1
```

ซึ่งในกรณีนี้ เครื่องเซิร์ฟเวอร์มีการ์ดแลนสองใบคือ eth0 และ eth1 และต้องการกำหนดไอพีของเครื่องเป็น 192.168.1.111 ให้มี gateway เป็น 192.168.1.1

4. จากนั้นให้รีบูตคอมพิวเตอร์

5. ตรวจสอบ

ใช้เทอร์มินัล

```
sut@SUTinsFEC5308:~$ ifconfig
```

```
bond0  Link encap:Ethernet HWaddr 00:22:2d:27:b1:83
        inet addr:192.168.1.111 Bcast:192.168.1.255 Mask:255.255.255.0
        inet6 addr: fe80::222:2dff:fe27:b183/64 Scope:Link
        UP BROADCAST RUNNING MASTER MULTICAST MTU:1500 Metric:1
        RX packets:62 errors:0 dropped:0 overruns:0 frame:0
        TX packets:112 errors:0 dropped:0 overruns:0 carrier:2
        collisions:0 txqueuelen:0
        RX bytes:7493 (7.4 KB) TX bytes:15189 (15.1 KB)
```

```
eth0   Link encap:Ethernet HWaddr 00:22:2d:27:b1:83
        inet addr:192.168.1.102 Bcast:192.168.1.255 Mask:255.255.255.0
        UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
        RX packets:52 errors:0 dropped:0 overruns:0 frame:0
        TX packets:109 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:6437 (6.4 KB) TX bytes:14740 (14.7 KB)
        Interrupt:19 Base address:0x8c00
```

```
eth1   Link encap:Ethernet HWaddr 00:22:2d:27:b1:83
        UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
        RX packets:10 errors:0 dropped:0 overruns:0 frame:0
        TX packets:3 errors:0 dropped:0 overruns:0 carrier:2
        collisions:0 txqueuelen:1000
        RX bytes:1056 (1.0 KB) TX bytes:449 (449.0 B)
        Interrupt:27
```

```
lo     Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
```



```

inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:37 errors:0 dropped:0 overruns:0 frame:0
TX packets:37 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:2170 (2.1 KB) TX bytes:2170 (2.1 KB)

```

จากข้อมูลจะเห็นว่า bond0 ใช้ไอพี 192.168.1.111 และเป็น Master ตามที่เรากำหนด
ข้อมูลของ eth1 ทุกอย่างถูกต้องคือไม่ได้ไอพีและเป็น Slave
ส่วน eth0 เนื่องจากทดลองกับ ADSL router มันจึงรับ ไอพีตามที่กำหนดไว้ด้วยโปรแกรม wicd
แต่ถึง eth0 จะได้อิพี 192.168.1.102 ก็ไม่มีปัญหาใด ถ้าต้องการออก Internet ให้ใช้คำสั่ง
sudo route del default eth0
เพื่อลบ routing ที่เกินทิ้งไป

6. ทดลอง

ใช้เทอร์มินัล

ให้ ping ไปที่ 192.168.1.111 ด้วยคำสั่ง

```
ping 192.168.1.111
```

จากนั้นที่เครื่องเซิร์ฟเวอร์ 192.168.1.111 ให้ใช้คำสั่ง

```
sut@SUTinsFEC5308:~$ ifconfig
```

```

bond0  Link encap:Ethernet HWaddr 00:22:2d:27:b1:83
        inet addr:192.168.1.111 Bcast:192.168.1.255 Mask:255.255.255.0
        inet6 addr: fe80::222:2dff:fe27:b183/64 Scope:Link
        UP BROADCAST RUNNING MASTER MULTICAST MTU:1500 Metric:1
        RX packets:773 errors:0 dropped:0 overruns:0 frame:0
        TX packets:436 errors:0 dropped:0 overruns:0 carrier:2
        collisions:0 txqueuelen:0
        RX bytes:64705 (64.7 KB) TX bytes:32045 (32.0 KB)

```

```

eth0  Link encap:Ethernet HWaddr 00:22:2d:27:b1:83
        inet addr:192.168.1.102 Bcast:192.168.1.255 Mask:255.255.255.0
        UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
        RX packets:389 errors:0 dropped:0 overruns:0 frame:0
        TX packets:369 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:34041 (34.0 KB) TX bytes:22890 (22.8 KB)
        Interrupt:19 Base address:0x8c00

```

```
eth1  Link encap:Ethernet HWaddr 00:22:2d:27:b1:83
      UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
      RX packets:384 errors:0 dropped:0 overruns:0 frame:0
      TX packets:67 errors:0 dropped:0 overruns:0 carrier:2
      collisions:0 txqueuelen:1000
      RX bytes:30664 (30.6 KB) TX bytes:9155 (9.1 KB)
      Interrupt:27

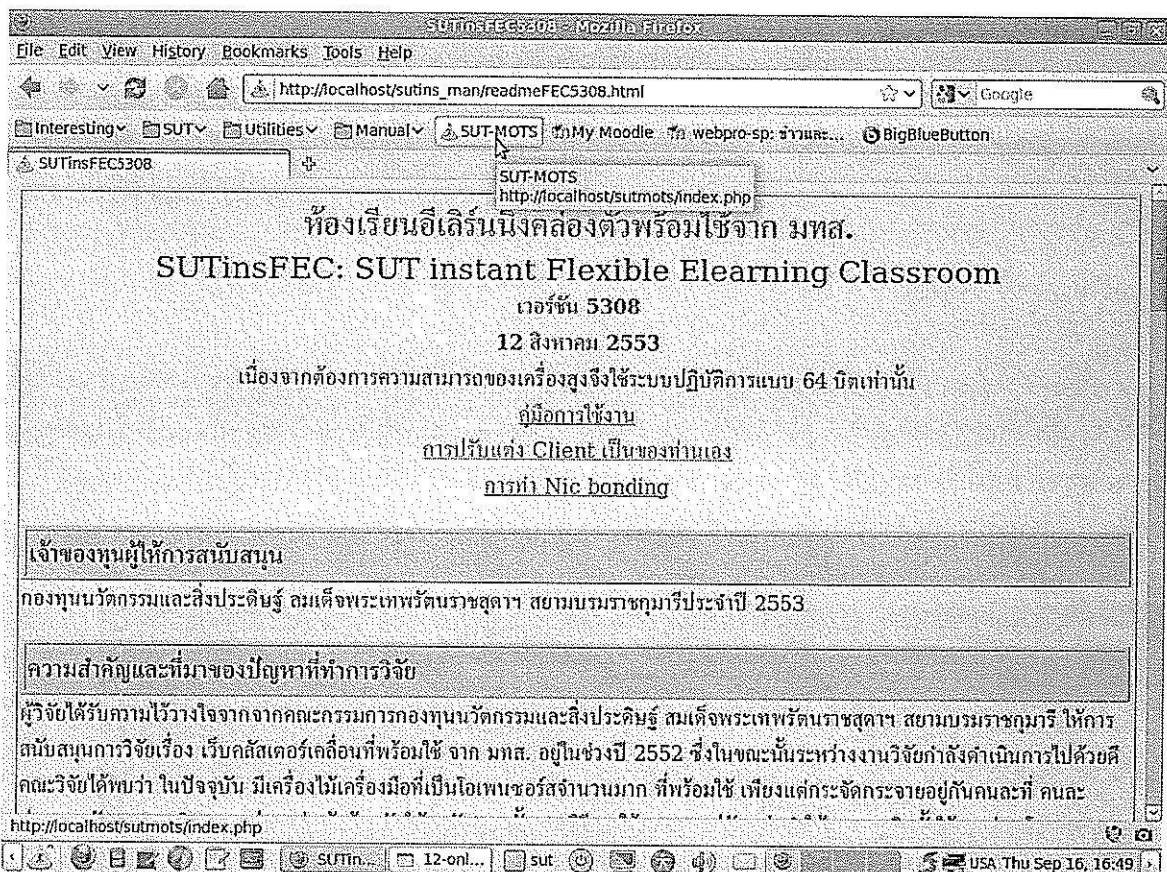
lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      RX packets:132 errors:0 dropped:0 overruns:0 frame:0
      TX packets:132 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:15520 (15.5 KB) TX bytes:15520 (15.5 KB)
```

จากข้อมูลตรงบรรทัดที่ขีดเส้นใต้ไว้ จะเห็นว่า มีการรับส่งข้อมูลของการ์ดแลนทั้งสองเกิดขึ้น

บทที่ 13

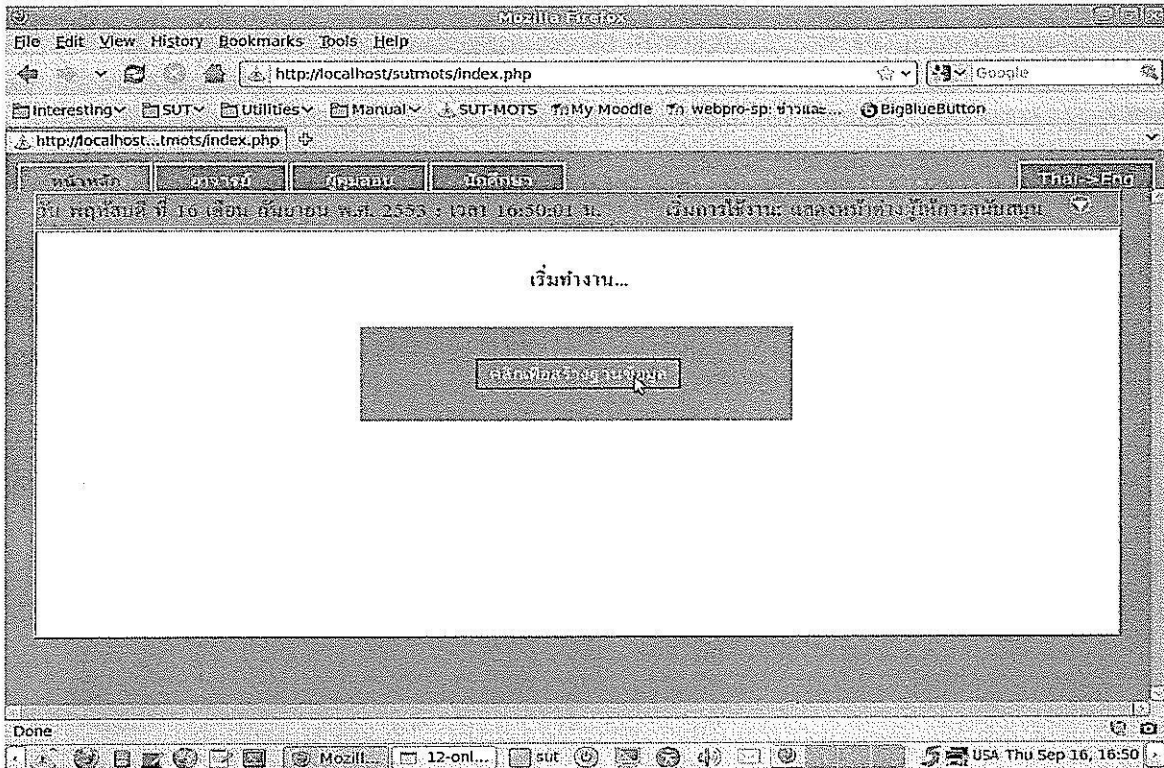
ระบบสอบออนไลน์เคลื่อนที่จาก มทส.

ระบบสอบออนไลน์เคลื่อนที่จาก มทส. เป็นงานวิจัยชิ้นหนึ่งของผู้วิจัยเอง ที่ได้รับทุนสนับสนุนการวิจัยจาก กองทุนนวัตกรรมและสิ่งประดิษฐ์สมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี ถึงสองครั้ง โดยครั้งแรกสำเร็จเมื่อวันที่ 5 ธันวาคม 2548 และครั้งที่สองสำเร็จเมื่อวันที่ 1 พฤศจิกายน 2550 ต่อจากนั้นได้มีการพัฒนาอย่างต่อเนื่องจนถึงรุ่น 5306 ซึ่งถือเป็นรุ่นที่ 13 ที่ได้บรรจุอยู่ใน SUTinsFEC5308 นี้ด้วย การเรียกใช้งานทำได้โดยคลิกที่ลิงก์ SUT-MOTS ที่อยู่บน Firefox ดังรูปที่ 13.1



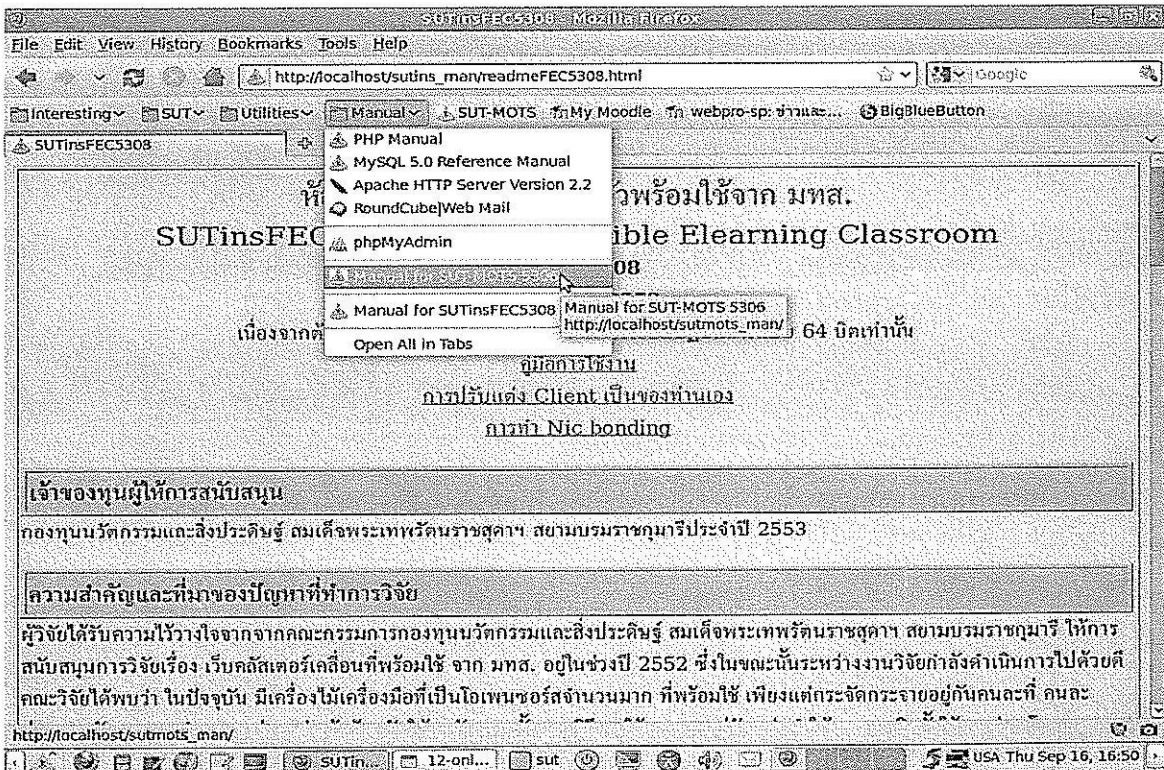
รูปที่ 13.1 ลิงก์เข้าสู่โปรแกรมระบบสอบออนไลน์

ก็จะเข้าสู่หน้าต่างของโปรแกรมระบบสอบออนไลน์ โดยเริ่มแรกจะมีการให้สร้างฐานข้อมูลก่อน ดังรูปที่ 13.2

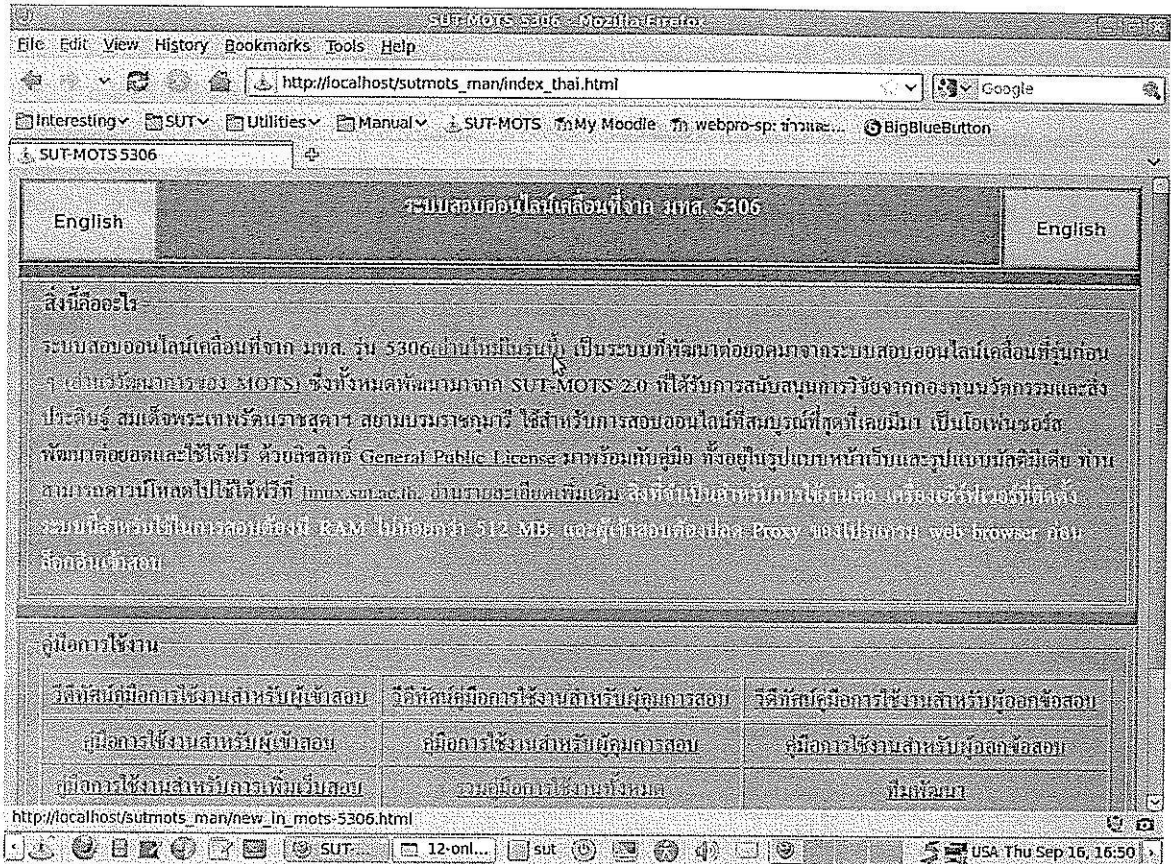


รูปที่ 13.2 เริ่มทำงานระบบสอบออนไลน์

เนื่องจากเป็นระบบที่มีโหมดการใช้งานถึงสามโหมดคือ โหมดอาจารย์ โหมดผู้คุมสอบและโหมดผู้เข้าสอบ และด้วยความสลับซับซ้อนของระบบ ดังนั้นจึงไม่สามารถกล่าวรายละเอียดได้มากนัก ในการวิจัยครั้งนี้ อย่างไรก็ตามผู้สนใจสามารถหารายละเอียดเพิ่มเติมได้จากคู่มือ ตามลิงก์ที่ปรากฏดังรูปที่ 13.3 และ 13.4



รูปที่ 13.3 อ่านคู่มือระบบสอบออนไลน์



รูปที่ 13.4 อานรายละเอียดของเวอร์ชัน 5306