

คำนำ

เอกสารฉบับนี้จัดทำจากเรียบเรียงความรู้การใช้ Linux ที่กระจายอยู่บนอินเทอร์เน็ต โดยมีจุดประสงค์เพื่อเผยแพร่การติดตั้งและใช้งาน Server ที่เป็น Ubuntu ในองค์กรต่างๆที่ไม่มีผู้เชี่ยวชาญคอยดูแลระบบให้ ดังนั้นผู้เรียบเรียงจึงเห็นว่าองค์กรเหล่านี้ต้องมีเอกสารเพื่อประกอบการติดตั้งและดูแลระบบเบื้องต้นไว้ในกรณีที่มีปัญหาและไม่สามารถเชื่อมต่อกับอินเทอร์เน็ตได้ รวมถึงต้องให้มีผู้ใช้งาน Sever ที่เป็น Linux เพิ่มมากขึ้นในประเทศเพื่อจะได้ลดค่าใช้จ่ายของซอฟต์แวร์ลงได้บ้าง

ขอขอบคุณ ผู้ที่เสียสละเขียนบทความและให้ความรู้ที่เกี่ยวข้อง ซึ่งหากขาดบุคคลเหล่านั้น เอกสารฉบับนี้คงจะเกิดขึ้นไม่ได้

วิรัช สว่างทุกข์
ผู้เรียบเรียง

แหล่งข้อมูลที่น่าสนใจในเอกสารฉบับนี้

<http://th.wikipedia.org/wiki/Linux>

<http://www.school.net.th/library/snet1/software/linux/index.html>

<http://course.rru.ac.th>

<http://linux.sothorn.org> โดย โสทร รอดคงที่

<http://www1.rbac.ac.th/techno/ubuntu/command.html>

http://www.thaicert.org/paper/basic/Secure_Shell.php โดย กิตติศักดิ์ จีรวรรณกุล

<http://opensource.cc.psu.ac.th> โดย วิภัทร ศรีติพรหม

รู้จักกับลินุกซ์.....6

โอเพนซอร์ซ.....	6
ประวัติลินุกซ์.....	7
การอ่านออกเสียง.....	8
การใช้งาน.....	8
การนำลินุกซ์มาใช้งาน.....	9

LINUX DISTRIBUTION.....9

รายชื่อ LINUX DISTRIBUTION ที่นิยมใช้งาน.....	10
---	----

พาร์ติชันฮาร์ดดิสก์.....11

การเรียกชื่อฮาร์ดดิสก์บนลินุกซ์.....	11
หลักการแบ่งพาร์ติชันเพื่อติดตั้งลินุกซ์ให้มีประสิทธิภาพ.....	12
การพาร์ติชันฮาร์ดดิสก์บน UBUNTU SERVER 8.10.....	13

การติดตั้ง UBUNTU SERVER.....14

ความต้องการของระบบ.....	14
ดาวน์โหลด UBUNTU SERVER.....	14
ขั้นตอนการติดตั้ง UBUNTU SERVER 8.10.....	14
ทำไมต้อง SUDO.....	32

คำสั่งพื้นฐานการใช้งานระบบปฏิบัติการ UNIX (LINUX).....33

คำสั่งเกี่ยวกับ FILE SYSTEM.....	33
การเปลี่ยน WORKING DIRECTORY.....	33
การแสดงรายชื่อ FILE และ DIRECTORY ย่อย.....	34
การจัดการ FILE.....	34
การจัดการเกี่ยวกับ DIRECTORY.....	35
คำสั่งแสดงข้อมูลใน FILE.....	36
คำสั่งที่ใช้นำจำนวนตัวอักษร จำนวนบรรทัด จำนวนคำใน FILE.....	36
คำสั่งกรองข้อมูล.....	37

คำสั่งใช้สำหรับค้นหา FILE.....	38
คำสั่งสำหรับสร้าง LINK.....	39
คำสั่ง ใช้สำหรับ FORMAT FILE SYSTEM	40
FILE COMPRESSION	41
คำสั่งเกี่ยวกับ PROCESS	42
คำสั่งที่ใช้แสดงรายละเอียดเกี่ยวกับ PROCESS	43
การใช้งาน vi เบื้องต้น.....	43
คำสั่งเพื่อใช้งาน FILE SYSTEM	44
FILE ที่เก็บค่าเริ่มต้นการทำงานของ LINUX ที่สำคัญ	46
FILE “/ETC/INITTAB”	46
FILE “/ETC/FSTAB”	47
FILE /ETC/HOSTS.....	47
FILE “/ETC/RESOLV.CONF”	47
FILE “/VAR/LOG/MESSAGES”	48
คำสั่ง บอกรายละเอียดของ HARDWARE ของระบบ	48
คำสั่งเกี่ยวกับ NETWORK.....	48
STARTUP และ SHUTDOWN INTERFACE.....	48
กำหนด IP ADDRESS ให้กับ INTERFACE	49
คำสั่งสำหรับแสดงสถานะการทำงานของ NETWORK INTERFACE CARD และรายละเอียดของ TCP/IP	49
คำสั่งตรวจสอบสถานะการติดต่อบน TCP/IP	50
คำสั่งตรวจสอบการส่งข้อมูลผ่าน ROUTER.....	50
การเปิดปิดเครื่อง SERVER.....	51
การจัดการ USER และ GROUP	52
การเพิ่ม GROUP	52
การลบ GROUP	52
การเพิ่ม USER.....	52
การกำหนด PASSWORD มีรูปแบบคำสั่งดังนี้	52
การลบ USER	53
FILE และ DIRECTORY PERMISSION	54
คำสั่งที่ใช้แสดงค่า PERMISSION ของ FILE หรือ DIRECTORY	54
ความหมายของตัวอักษรใน PERMISSION.....	54
คำสั่งที่ใช้กำหนด PERMISSION	55
<u>การจัดการแพ็คเกจบน UBUNTU</u>	<u>56</u>
การติดตั้งแพ็คเกจแบบออฟไลน์ด้วย DPKG.....	56

การติดตั้งแพ็คเกจแบบออนไลน์ ด้วย APT-GET.....	57
การติดตั้งแพ็คเกจแบบออนไลน์ด้วย APTITUDE	59
APT-GET หรือ APTITUDE ผ่าน PROXY AUTHENTICATION.....	60

การ CONFIG SERVER ต่างๆ บน UBUNTU61

การกำหนด IP ADDRESS	61
หากต้องการ SETUP INTERFACE เป็น DHCP ทำการแก้ไขดังนี้	61
การกำหนด DNS SERVER	61
การทำ IP FORWARD	62
การติดตั้งและคอนฟิกเนมเซิร์ฟเวอร์ในองค์กร.....	63
การติดตั้ง BIND9 บน UBUNTU SERVER	63
เว็บเซิร์ฟเวอร์ในองค์กร.....	71
การติดตั้ง LAMP SERVER	72
DYNAMIC HOST CONFIGURATION PROTOCOL.....	76
การติดตั้ง DHCP.....	77
PROXY SERVER คืออะไร.....	79
ติดตั้ง SQUID	79
การใช้งาน PROXY เป็น TRANSPARENT PROXY	80
การใช้ SECURE SHELL ในการเข้าถึงระบบจากระยะไกล	81

รู้จักกับลินุกซ์

โอเพนซอร์ซ

โอเพนซอร์ซ หรือ โอเพนซอร์ส (open source) คือวิธีการในการออกแบบ พัฒนา และแจกจ่าย สำหรับต้นฉบับของสินค้าหรือความรู้ โดยเฉพาะซอฟต์แวร์ โดยโอเพนซอร์ซถูกพิจารณาว่าเป็นทั้งรูปแบบหนึ่งในการออกแบบ และแผนการในการดำเนินการ โดยโอเพนซอร์ซเปิดโอกาสให้บุคคลอื่นนำเอาระบบนั้นไปพัฒนาได้ต่อไป นอกเหนือจากด้านซอฟต์แวร์คำว่าโอเพนซอร์ซยังคงเริ่มนำมาใช้ในส่วนของไบโอเทคโนโลยี

ซอฟต์แวร์โอเพนซอร์ซ

ซอฟต์แวร์โอเพนซอร์ซ เริ่มต้นจากการเคลื่อนไหวภายใต้ชื่อซอฟต์แวร์เสรี (free software) ในช่วง พ.ศ. 2526 จนกระทั่งในปี 2531 คำว่าซอฟต์แวร์โอเพนซอร์ซได้ถูกนำมาใช้แทนคำว่า "ฟรี" เพื่อให้ง่ายต่อการเข้าใจและให้ความรู้สึกสบายใจต่อทั้งผู้ใช้และผู้พัฒนา รวมถึงคำว่า ฟรี ในลักษณะของคำว่าเสรีนอกเหนือจากคำว่าฟรีในลักษณะไม่เสียค่าใช้จ่าย ผู้ใช้งานรวมถึงผู้พัฒนาสามารถนำซอฟต์แวร์มาใช้งาน แก้ไข แจกจ่าย โดยสามารถนำมาปรับปรุงทั้งในลักษณะส่วนตัว หรือในหน่วยงานเอกชนได้ ซอฟต์แวร์โอเพนซอร์ซอนุญาตให้ทุกคนสามารถนำซอฟต์แวร์ไปพัฒนา รวมถึงวางขายและทำการตลาด ซอฟต์แวร์โอเพนซอร์ซที่เป็นที่รู้จักกันดีได้แก่ เฟิร์ล, ไฟร์ฟอกซ์, ลินุกซ์, อะแพชี เว็บเซิร์ฟเวอร์ ลักษณะเงื่อนไขทางลิขสิทธิ์ที่นิยมได้ สัญญาอนุญาตสาธารณะทั่วไปของกนู (จีพีแอล) และ สัญญาอนุญาตแจกจ่ายซอฟต์แวร์ของเบิร์กลีย์ (บีเอสดี) จากรายงานของกลุ่มสแตนดิชประมาณการประหยัดงบประมาณจากการใช้งานซอฟต์แวร์โอเพนซอร์ซได้ 6 หมื่นล้านเหรียญสหรัฐต่อปี

ลินุกซ์ (Linux) และรู้จักในชื่อ กนู/ลินุกซ์ (GNU/Linux) โดยทั่วไปเป็นคำที่ใช้ในความหมายที่หมายถึงระบบปฏิบัติการแบบยูนิกซ์ โดยใช้ลินุกซ์ เคอร์เนล เป็นศูนย์กลางทำงานร่วมกับไลบรารีและเครื่องมืออื่น ลินุกซ์เป็นตัวอย่างหนึ่งในฐานะซอฟต์แวร์เสรี และซอฟต์แวร์โอเพนซอร์ซที่ประสบความสำเร็จและมีชื่อเสียง ทุกคนสามารถดูหรือนำโค้ดของลินุกซ์ไปใช้งาน, แก้ไข, และแจกจ่ายได้อย่างเสรี ลินุกซ์นิยมจำหน่ายหรือแจกฟรีในลักษณะเป็นแพคเกจ โดยผู้จัดทำจะรวมซอฟต์แวร์สำหรับใช้งานในด้านอื่นเป็นชุดเข้าด้วยกัน

เริ่มแรกของของลินุกซ์พัฒนาและใช้งานในเฉพาะกลุ่มผู้ที่สนใจ ซึ่งในปัจจุบันลินุกซ์ได้รับความนิยมเนื่องมาจากระบบการทำงานที่เป็นอิสระ ปลอดภัย เชื่อถือได้ และราคาต่ำ จึงได้มีการพัฒนาจากองค์กรต่าง ๆ เช่น ไอบีเอ็ม ฮิวเลตต์-แพคการ์ด และ โนวอลล์ ใช้สำหรับในระบบเซิร์ฟเวอร์และพีซี เริ่มแรกลินุกซ์พัฒนาสำหรับใช้กับเครื่อง อินเทล 386 ไมโครโพรเซสเซอร์ หลังจากที่ได้รับคามนิยมปัจจุบัน ลินุกซ์ได้พัฒนารับรองการใช้งานของระบบสถาปัตยกรรมคอมพิวเตอร์ในระบบต่าง ๆ รวมถึงในโทรศัพท์มือถือ และกล้องวิดีโอ

ลินุกซ์มีสัญญาอนุญาตแบบ GPL ซึ่งเป็นสัญญาอนุญาตที่กำหนดให้ผู้ทำนำโค้ดไปใช้ต้องใช้สัญญาอนุญาตแบบเดิม ต่อคือใช้สัญญาอนุญาต GPL เช่นเดียวกัน ซึ่งลักษณะสัญญาอนุญาตแบบนี้เรียกว่า copyleft

ประวัติลินุกซ์

ผู้เริ่มพัฒนาลินุกซ์ เคอร์เนลเป็นคนแรก คือ ลินุส โตร์วัลดส์ (Linus Torvalds) ชาวฟินแลนด์ เมื่อสมัยที่เขายังเป็นนักศึกษาคอมพิวเตอร์ ที่มหาวิทยาลัยเฮลซิงกิ โดยแรกเริ่ม ริชาร์ด สตอลแมน (Richard Stallman) ได้ก่อตั้งโครงการกนูขึ้นในปี พ.ศ. 2526 จุดมุ่งหมายโครงการกนู คือ ต้องการพัฒนาระบบปฏิบัติการคล้ายยูนิกซ์ที่เป็นซอฟต์แวร์เสรีทั้งระบบ ราวช่วงพ.ศ. 2533 โครงการกนูมีส่วนโปรแกรมที่จำเป็นสำหรับระบบปฏิบัติการเกือบครบทั้งหมด ได้แก่ คลังโปรแกรม (Libraries) คอมไพเลอร์ (Compiler) โปรแกรมแก้ไขข้อความ(Text Editor) และเปลือกระบบยูนิกซ์(Shell) ซึ่งขาดแต่เพียงเคอร์เนล(Kernel) เท่านั้น ในพ.ศ. 2533 โครงการกนูได้พัฒนาเคอร์เนลชื่อ Hurd เพื่อใช้ในระบบกนู ซึ่งในขณะนั้นมีปัญหาเกี่ยวกับความเร็วในการประมวลผล

ในพ.ศ. 2534 โตร์วัลดส์เริ่มโครงการพัฒนาเคอร์เนล ขณะศึกษาในมหาวิทยาลัยแล้ว โดยอาศัย Minix ซึ่งเป็นระบบที่คล้ายกับ Unix ซึ่งมากับหนังสือเรื่องการออกแบบระบบปฏิบัติการ มาเป็นเป็นต้นแบบในการเขียนขึ้นมาใหม่โดย Torvalds เขาพัฒนาโดยใช้ IA-32 assembler และภาษาซี คอมไพล์เป็นไฟล์ไบนารีและบูทจากแผ่นฟลอปปี้ดิสก์ เขาได้พัฒนามาเรื่อยๆจนกระทั่งสามารถบูทตัวเองได้ (กล่าวคือสามารถคอมไพล์ภายในลินุกซ์ได้เลย) และในปัจจุบันมีนักพัฒนาจากพันกว่าคนทั่วโลกได้เข้ามามีส่วนรวมในการพัฒนา โครงการ Eric S. Raymond ได้ศึกษากระบวนการพัฒนาดังกล่าวและเขียนบทความเรื่อง The Cathedral and the Bazaar

ในรุ่น 0.01 นี้ถือว่ามีเครื่องมือที่เพียงพอสำหรับระบบ POSIX ที่ใช้เรียก ลินุกซ์ ที่รันกับ กนู Bash Shell และมีการพัฒนาอย่างต่อเนื่องและอย่างรวดเร็ว

โตร์วัลดส์ยังคงมุ่งมั่นพัฒนาระบบต่อไป ซึ่งต่อมาก็สามารถรันบน X Window System และมีการเลือกนกเพนกวินที่ชื่อ Tux ให้เป็นตัวนำโชคหรือ Mascot ของระบบลินุกซ์



ริชาร์ด สตอลแมน ผู้ก่อตั้ง โครงการกนู,



ลินุส โตร์วัลดส์ ผู้ให้กำเนิด ลินุกซ์ เคอร์เนล

การอ่านออกเสียง

ในขณะที่คนไทยใช้คำว่า "ลินุกซ์" ในหลายประเทศที่ใช้ภาษาอังกฤษเป็นหลักนิยมออกเสียงเป็น "ลินิกซ์" หรือ "โลนิกซ์" โดยพยางค์ท้ายอ่านเหมือนพยางค์ท้ายของคำว่า "ยูนิกซ์" โดยลินุส โตร์วัลดส์ผู้คิดค้นลินุกซ์ได้กล่าวไว้ว่า "li" อ่านเหมือนเสียงสระอิ /I/ และ "nux" อ่านเสียงสระเหมือนเสียง /U/ ซึ่งคล้ายเสียง "อุ" ในภาษาไทย ในอินเทอร์เน็ตมีไฟล์ที่เก็บคำพูดของ โตร์วัลดส์ ที่พูดว่า "Hello, this is Linus Torvalds, and I pronounce Linux as Linux"^[1] เก็บไว้เนื่องจากมีการโต้เถียงกันมากในเรื่องการออกเสียง แม้ว่าจะมีการออกเสียงในแบบต่าง ๆ มากมาย โตร์วัลดส์ก็เคยกล่าวไว้ว่า การอ่านออกเสียงถูกหรือผิดไม่สำคัญ เพราะทุกคนเข้าใจตรงกันว่าหมายถึงอะไร

การใช้งาน

การใช้งานดั้งเดิมของลินุกซ์ คือ ใช้เป็นระบบปฏิบัติการสำหรับเครื่องเซิร์ฟเวอร์ แต่จากราคาที่ต่ำ ความยืดหยุ่น พื้นฐานจากยูนิกซ์ ทำให้ลินุกซ์เหมาะกับงานหลาย ๆ ประเภท

ลินุกซ์ ถือเป็นส่วนสำคัญของซอฟต์แวร์เซิร์ฟเวอร์ที่เรียกว่า LAMP ย่อมาจาก Linux, Apache, MySQL, Perl/PHP/Python ซึ่งเป็นที่นิยมใช้เป็นเว็บเซิร์ฟเวอร์ และพบมากสุดระบบหนึ่ง ตัวอย่างซอฟต์แวร์ซึ่งพัฒนาสำหรับระบบนี้คือ มีเดียวิกิ ซอฟต์แวร์สำหรับวิกิพีเดีย

เนื่องจากราคาที่ต่ำและการปรับแต่งได้หลากหลาย ลินุกซ์ถูกนำมาใช้ในระบบฝังตัว เช่น เครื่องรับสัญญาณโทรทัศน์ โทรศัพท์มือถือ และอุปกรณ์พกพาต่าง ๆ ลินุกซ์เป็นคู่แข่งที่สำคัญของ ซิมเบียนโอเอส ซึ่งใช้ในโทรศัพท์มือถือจำนวนมาก และใช้แทนวินโดวส์ซีอี และปาล์มโอเอส บนเครื่องคอมพิวเตอร์พกพา เครื่องบันทึกวีดิโอก็ใช้ลินุกซ์ที่ดัดแปลงเป็นพิเศษ ไฟร์วอลล์และเราเตอร์หลายรุ่น เช่น ของ Linksys ใช้ลินุกซ์และขีดความสามารถเรื่องทางเครือข่ายของมัน

ระยะหลังมีการใช้ลินุกซ์เป็นระบบปฏิบัติการของซูเปอร์คอมพิวเตอร์มากขึ้น ในรายชื่อซูเปอร์คอมพิวเตอร์ TOP500 ของเดือนพฤศจิกายน พ.ศ. 2548 เครื่องซูเปอร์คอมพิวเตอร์ที่เร็วที่สุดสองอันดับแรกใช้ลินุกซ์ และจาก 500 ระบบ มีถึง 371 ระบบ (คิดเป็น 74.2%) ให้ลินุกซ์แบบใดแบบหนึ่ง

เครื่องเล่นเกม โซนี่ เพลย์สเตชัน 3 ที่ออกในปี พ.ศ. 2549 รันลินุกซ์ โซนี่ยังได้ปล่อย PS2 Linux สำหรับใช้กับเพลย์สเตชัน 2 อีกด้วย ผู้พัฒนาเกมอย่าง Atari และ id Software ก็เคยออกซอฟต์แวร์เกมบนลินุกซ์มาแล้ว (อ้างอิงจาก<http://th.wikipedia.org/wiki/Linux>)

ลินุกซ์ เป็นระบบปฏิบัติการเช่นเดียวกับ ดอส ไมโครซอฟต์วินโดวส์ หรือยูนิกซ์ โดยลินุกซ์นั้นจัดว่าเป็นระบบปฏิบัติการยูนิกซ์ประเภทหนึ่ง การที่ลินุกซ์เป็นที่กล่าวขานกันมากขณะนี้ เนื่องจากความสามารถของตัวระบบปฏิบัติการและโปรแกรมประยุกต์ที่ทำงานบนระบบ ลินุกซ์ โดยเฉพาะอย่างยิ่งโปรแกรมในตระกูลของ GNU (GNU's Not UNIX) และสิ่งที่สำคัญที่สุดก็คือระบบลินุกซ์เป็นระบบปฏิบัติการประเภทฟรีแวร์ (Free Ware) คือไม่เสียค่าใช้จ่ายในการซื้อโปรแกรม

ระบบลินุกซ์ตั้งแต่ เวอร์ชัน 4 นั้น สามารถทำงานได้บนชิพซีพียูทั้ง 3 ตระกูล คือบนชิพซีพียูของอินเทล (PC Intel) ดิจิตอลอัลฟาคอมพิวเตอร์ (Digital Alpha Computer) และซันสปาร์ก (SUN SPARC) เนื่องจากใช้เทคโนโลยีที่เรียกว่า RPM (Red Hat Package Management) ถึงแม้ว่าในขณะนี้ลินุกซ์ยังไม่สามารถแทนที่ไมโครซอฟต์ วินโดวส์ บนพีซีหรือแมคโอเอส (Mac OS) ได้ทั้งหมดก็ตาม แต่ผู้ใช้จำนวนไม่น้อยที่หันมาใช้และช่วยพัฒนาโปรแกรมประยุกต์บนลินุกซ์กัน และเรื่องของการดูแลระบบลินุกซ์นั้น ภายในระบบลินุกซ์เองมีเครื่องมือช่วยสำหรับดำเนินการให้สะดวกยิ่งขึ้น

การนำลินุกซ์มาใช้งาน

ปัจจุบัน ได้มีการนำระบบปฏิบัติการลินุกซ์ไปประยุกต์เป็นระบบปฏิบัติการสำหรับงานด้านต่างๆ เช่นงานด้านการคำนวณทางวิทยาศาสตร์ใช้เป็นสถานีนงาน สถานีบริการ อินเทอร์เน็ต อินทราเน็ต หรือใช้ใน การเรียนการสอนและการทำวิจัยทางคอมพิวเตอร์ใช้พัฒนาโปรแกรมเนื่องจาก มีเครื่องมือมากมาย เช่น โปรแกรมภาษาซี (C) ซีพลัสพลัส (C++) ปาสคาล (Pascal) ฟอรัทเรน (Fortran) ลิสป์ (Lisp) โปรล็อก (Prolog) เอดา (ADA) มีภาษาสคริปต์ เช่น เชลล์ (Shell) บาสซ์เชลล์ (Bash Shell) ซีเชลล์ (C Shell) คอร์นเชลล์ (Korn Shell) เพิร์ล (Perl) พายตัน (python) TCL/TK นอกจากนี้ยังมีโปรแกรมประยุกต์ในสาขาต่างๆ อีกมากมาย โดยข้อมูลของโปรแกรมเหล่านี้ได้รวบรวมไว้ที่ Linux Software Map (LSM) (อ้างอิงจาก <http://www.school.net.th/library/snet1/software/linux/index.html>)

Linux Distribution

ในยุคแรกที่มีการนำ Linux มาใช้งาน มักจะเป็นกลุ่มนักคอมพิวเตอร์ที่มีความเชี่ยวชาญสูง โดยจะต้องดาวน์โหลดส่วนประกอบของ Linux มาจากอินเทอร์เน็ต ได้แก่ Kernel ,Shell ,C Compiler และ Boot Loader จากนั้นจะต้องนำส่วนประกอบทั้งหลายมาคอมไพล์ และติดตั้งทีละส่วน จนกระทั่งเป็นระบบปฏิบัติการที่สามารถใช้งานได้ ตามต้องการ สร้างความยากลำบาก และใช้เวลามากกว่าจะการติดตั้งจะสำเร็จลงได้

จากปัญหาข้างต้น จึงทำให้มีบริษัทซอฟต์แวร์ที่เรียกว่า Linux Distribution เป็นผู้ผลิตระบบปฏิบัติการขึ้นเป็นของตนเอง โดยนำเอา Kernel ของ Linux มาประกอบเข้ากับซอฟต์แวร์อื่น ๆ และจัดทำให้เป็นผลิตภัณฑ์สำเร็จรูป ความแตกต่างของ Linux Distribution แต่ละรายแบ่งออกได้ดังนี้

การติดตั้ง ขั้นตอนของการติดตั้งมีตั้งแต่วิธีการแบบแมนนวล คือ ผู้ติดตั้งกระทำเองทุกขั้นตอน ไปจนถึง ติดตั้งแบบอัตโนมัติทุกอย่าง บางรายมีเมนูอธิบายขั้นตอนการติดตั้งเป็นภาษาไทยก็มี เช่น Ubuntu ,Mandrake 8.x ,Fedora

ความง่ายในการตั้งค่าให้แก่ระบบ Linux Distribution แต่ละรายจะมีเครื่องมือที่ใช้ในการคอนฟิกระบบแตกต่างกันไป บางรายอาจจะมีเครื่องมือช่วยการคอนฟิกแบบรวมศูนย์กลาง เช่น YaST ของ SUSE Linux apt ของ Debian ,Ubuntu yum ของ Fedora เป็นต้น

ซอฟต์แวร์ที่แถมให้มาพร้อมกัน บาง Distribution อาจจะทำให้ซอฟต์แวร์แบบฟรีแวร์มานับเป็นพัน ๆ โปรแกรม แต่บางรายอาจจะทำให้ซอฟต์แวร์เฉพาะมาเท่าที่จำเป็นจริง ๆ เท่านั้น

ระบบ X Window ส่วนมากแล้ว ทุก ๆ Distribution จะมีระบบ X Window ซึ่งเป็น GUI ของระบบยูนิกซ์ให้มาพร้อมกันด้วย ซึ่งนิยมใช้โปรเจก XFree86 ดังนั้นจึงมีลักษณะการติดตั้งและใช้งานที่คล้ายคลึงกัน

การสนับสนุนทางเทคนิคหลังการขาย สำหรับองค์กรขนาดใหญ่ การให้การสนับสนุนทางเทคนิคมีความจำเป็นมาก หลาย ๆ distribution มักจะแบ่งจำหน่ายสินค้าของตนเองทั้งแบบดาวน์โหลดฟรี (ไม่มีการสนับสนุนทางเทคนิค) แบบมีการสนับสนุนทางเทคนิคระยะเวลาหนึ่ง และแบบทำสัญญาเป็นระยะยาว

Linux Distribution ที่ได้รับความนิยมสูงในประเทศไทย ได้แก่ Ubuntu ,Fedara ,Slackware ,Redhat และ Mandrake

Linux Distribution ที่พัฒนาขึ้นโดยคนไทย ได้แก่ Kaiwal , ZiFF , SIS , Linux TLE 4.0

Linux Distribution อื่น ๆ ได้แก่ Caldera OpenLinux , Debian GNU/Linux , S.u.S.E

รายชื่อ *Linux Distribution* ที่นิยมใช้งาน

Ubuntu

Debian

Slackware

RedHat

Caldelra

SLS (Softlanding Linux System)

Yggdrasil

Mandrak

Fedora

SuSE

TLE

Lindows

Liberta

พาร์ติชันฮาร์ดดิสก์

การพาร์ติชันฮาร์ดดิสก์ คือการจัดแบ่งพื้นที่ของฮาร์ดดิสก์เพื่อเอามาใช้งานอย่างเหมาะสม และเพื่อความสะดวกในการเก็บข้อมูล สำหรับลินุกซ์การแบ่งพาร์ติชันมีข้อดีหลายอย่าง เช่น รักษาความปลอดภัยของเครื่อง สามารถทำไคลด์ได้ สะดวกในการสำรองข้อมูล ฯลฯ

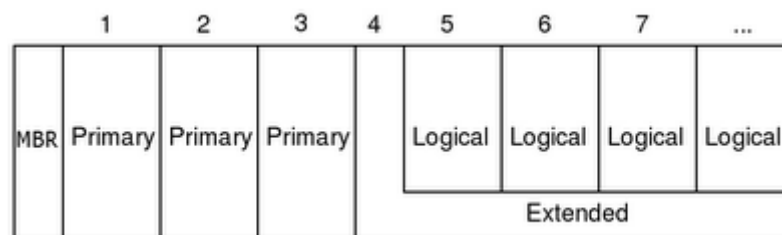
การสร้างพาร์ติชัน ให้กับฮาร์ดดิสก์ มีอยู่ 3 แบบ ได้แก่

1.Primary Partition

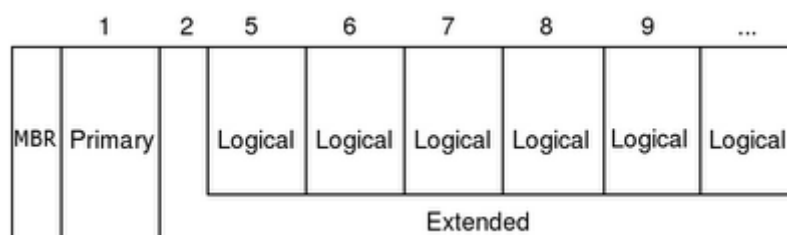
2.Extended Partition

3.Logical Partition

ในการสร้างพาร์ติชันฮาร์ดดิสก์ ถ้าเราสร้างทุกพาร์ติชันให้เป็น Primary ทั้งหมด จะสร้างได้เพียง 4 พาร์ติชันเท่านั้น ถ้าต้องการมากกว่านั้นต้องใช้ 1 พาร์ติชันเป็น Extended แล้ว แบ่งย่อย Extended เป็น Logical ดังภาพ



อีกตัวอย่าง



พาร์ติชันแรกของ Logical partition จะเป็นพาร์ติชัน ที่ 5 เสมอ

ลินุกซ์เคอร์เนลมีข้อจำกัดในการจัดการพาร์ติชันที่เป็นฮาร์ดดิสก์ IDE ได้ 63 พาร์ติชัน ฮาร์ดดิสก์ SCSI จะได้ 15 พาร์ติชัน

การเรียกชื่อฮาร์ดดิสก์บนลินุกซ์

ฮาร์ดดิสก์ IDE มีการเชื่อมต่อกับสาย IDE ตรงตำแหน่งต่างๆ จะมีชื่อเรียกดังนี้

Primary Master เรียกว่า /dev/hda

Primary Slave เรียกว่า /dev/hdb
 Secondary Master เรียกว่า /dev/hdc
 Secondary Slave เรียกว่า /dev/hdd
 ลำดับที่ของ พาร์ติชันของฮาร์ดิสก์ลำดับที่เท่าไรก็จะเรียก /dev/hda1, /dev/hda2 ..., /dev/hdb1, /dev/hdb2..., /dev/hdc1, /dev/hdc2...

ฮาร์ดิสก์ SCSI จะเรียกชื่อตาม SCSI ID

SCSI ID 0 เรียกว่า /dev/sda
 SCSI ID 1 เรียกว่า /dev/sdb
 SCSI ID 2 เรียกว่า /dev/sdc
 SCSI ID 3 เรียกว่า /dev/sdd
 SCSI ID.. เรียกว่า /dev/sd...

เรื่อยไปตามจำนวนฮาร์ดิสก์ที่สามารถใส่ได้ของ SCSI ลำดับที่ของพาร์ติชันก็เช่นเดียวกันกับฮาร์ดิสก์แบบ IDE เช่น /dev/sda1, /dev/sda2..., /dev/sdb1, /dev/sdb2..., /dev/sdc1, /dev/sdc2... ส่วนฮาร์ดิสก์ SATA ก็จะใช้ชื่อ พาร์ติชันเหมือนกับฮาร์ดิสก์ SCSI

หลักการแบ่งพาร์ติชันเพื่อติดตั้งลินุกซ์ให้มีประสิทธิภาพ

(อ้างอิงจาก linux.sothorn.org)

ความเป็นจริงแล้วการแบ่งพาร์ติชันเพื่อการติดตั้งลินุกซ์นั้น ถ้าแบ่งพาร์ติชันเพื่อให้ติดตั้งได้ก็ไม่มีอะไรมาก แค่มีพาร์ติชัน / (รูท) และ swap ก็เพียงพอแล้ว แต่ถ้านำไปใช้เป็นเซิร์ฟเวอร์ใช้งานจริง จะมีปัญหาต่างๆ ตามมามาก เช่น ไม่สามารถกำหนดโควต้าให้กับผู้ใช้งานได้ ถ้าเกิดข้อมูลใน / (รูท) เต็มคราวนี้เซิร์ฟเวอร์บางอย่างก็สตาร์ทไม่ขึ้น เกิดถูกแฮก แล้วแฮกเกอร์ไปแก้ไฟล์ใน /boot ตอนนี้อยู่ดูแลระบบคงนึกภาพออกว่ามันโหดร้ายขนาดไหน

การแบ่งพาร์ติชันไม่ได้มีหลักการตายตัว แต่ผู้ติดตั้งต้องรู้ว่าจะติดตั้งเซิร์ฟเวอร์เพื่อเอาไปใช้ประโยชน์อะไร และที่เก็บข้อมูลอยู่ตรงไหน เช่น

ข้อมูลของผู้ใช้ อยู่ใน /home
 จดหมายเข้าอยู่ที่ /var/mail
 ที่เก็บข้อมูลของเว็บอยู่ที่ /var/www
 ที่เก็บข้อมูลของ MySQL อยู่ที่ /var/lib/mysql

ถ้ารู้แบบนี้สามารถแบ่งพาร์ติชันได้อย่างมีประสิทธิภาพแล้ว

ตัวอย่าง เซิร์ฟเวอร์

/boot	100 MB	พื้นที่เก็บ Kernel และ ไฟล์ที่เกี่ยวข้องกับการบูท
/	10 GB	พื้นที่ของลินุกซ์ซิสเต็ม
/home	5 GB	พื้นที่เก็บข้อมูลของผู้ใช้งานทุกคน
/backup	5 GB	พื้นที่ให้แอดมินไว้แบคอัพ
/tmp	256 MB	พื้นที่เก็บไฟล์ชั่วคราว
swap	1 GB	พื้นที่ที่ใช้เป็นหน่วยความจำสำรอง เวลา RAM ไม่พอ ตั้งให้มีขนาด

2 เท่าของ RAM แต่ไม่เกิน 2 GB

มาดูข้อดีของเซิร์ฟเวอร์เครื่องนี้

1. แบ่ง พาร์ติชัน /boot ออกมา สามารถกำหนด ให้เป็นพาร์ติชันที่อ่านได้อย่างเดียวในไฟล์ /etc/fstab สมมติว่าถูกแฮก แฮกเกอร์ก็ไม่สามารถลบหรือแก้ไขไฟล์ ใน /boot ได้
2. สามารถทำโควต้ากำหนดพื้นที่ใช้งานให้กับผู้ใช้ใน /home ได้
3. พื้นที่ /backup ไม่เกี่ยวข้องกับ /
4. ถ้าเกิดนำข้อมูลมาใส่ใน / จนเต็ม ก็ไม่กระทบกับระบบ เพราะ /tmp ออกมาต่างหากแล้ว

การพาร์ติชันฮาร์ดดิสก์บน *Ubuntu Server 8.10*

จากเรื่อง หลักการแบ่งพาร์ติชันเพื่อติดตั้งลินุกซ์ให้มีประสิทธิภาพ ท่านคงทราบหลักการคร่าวๆ สำหรับการแบ่งพาร์ติชันไปแล้ว คราวนี้เรามาคุยกันว่าในขั้นตอนติดตั้ง Ubuntu Server จะแบ่งพาร์ติชันอย่างไร ระบบติดตั้งของ Ubuntu Server จะไม่มีระบบติดตั้งแบบ GUI หรือเมื่อติดตั้งเสร็จ Ubuntu Server ก็ไม่มี GUI มาให้ (ติดตั้งได้ภายหลังหากต้องการ) ท่านที่เคยชินอยู่กับลินุกซ์ตระกูล Red Hat อาจจะไม่เคยชินหรือไม่ชอบ เพราะเมื่อระบบมีปัญหาผู้ดูแลระบบต้องแก้ปัญหาโดยการใช้เวลาน้อยที่สุด ติดตั้งก็ต้องเร็ว และเซิร์ฟเวอร์ไม่ต้องเสีย RAM ให้กับระบบ GUI

การติดตั้ง Ubuntu Server

ความต้องการของระบบ

Ubuntu 8.10 รองรับการทำงาน บน CPU Intel x86 และ AMD64

CPU ขั้นต่ำ 300 MHz RAM 128 MB

พื้นที่ฮาร์ดดิสก์ 500 MB สำหรับติดตั้งระบบพื้นฐาน 1 GB สำหรับติดตั้งระบบทั้งหมด

ดาวน์โหลด Ubuntu Server

สามารถดาวน์โหลด Ubuntu Server ได้ที่ <http://www.ubuntu.com/getubuntu/download> ไฟล์ที่ดาวน์โหลดมาชื่อไฟล์ ubuntu-8.10.1-server-amd64.iso หรือ ubuntu-8.10.1-server-i386.iso นำไฟล์เหล่านี้มาเขียนลง CD ด้วยโปรแกรมเขียน CD เช่น Nero ด้วยเมนู Burn CD Image เมื่อได้แผ่นมาแล้วก็สามารถนำมาติดตั้งได้เลย

ขั้นตอนการติดตั้ง Ubuntu Server 8.10

1. ตั้งค่า Bios ให้บูทจาก CD-Rom เป็นอันดับแรก พร้อมแล้วก็ใส่ CD แล้ว รีบูทเครื่องใหม่ เมื่อบูทด้วย CD-ROM แล้ว จะปรากฏหน้าจอแรก คือให้เลือกภาษาที่จะใช้ในการติดตั้ง (เลือกภาษาอังกฤษ)



2. เมื่อเลือกภาษาเรียบร้อยแล้วก็มี Option ต่างให้เลือก โดยการ กด F1 F2 F3... ตามต้องการ



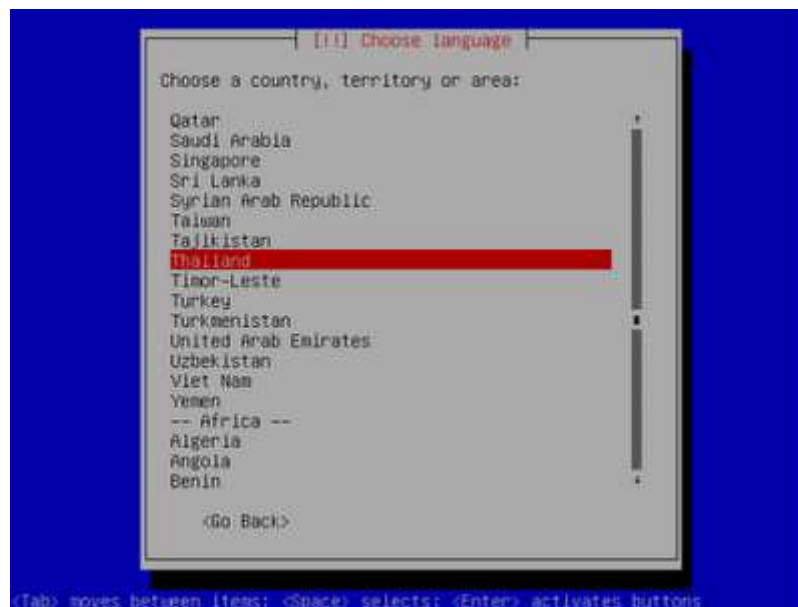
3. ให้เลือกภาษาที่ใช้สำหรับการติดตั้งอีกครั้ง



4. เลือกภาษาหากไม่มีภาษาไทย เลือก other

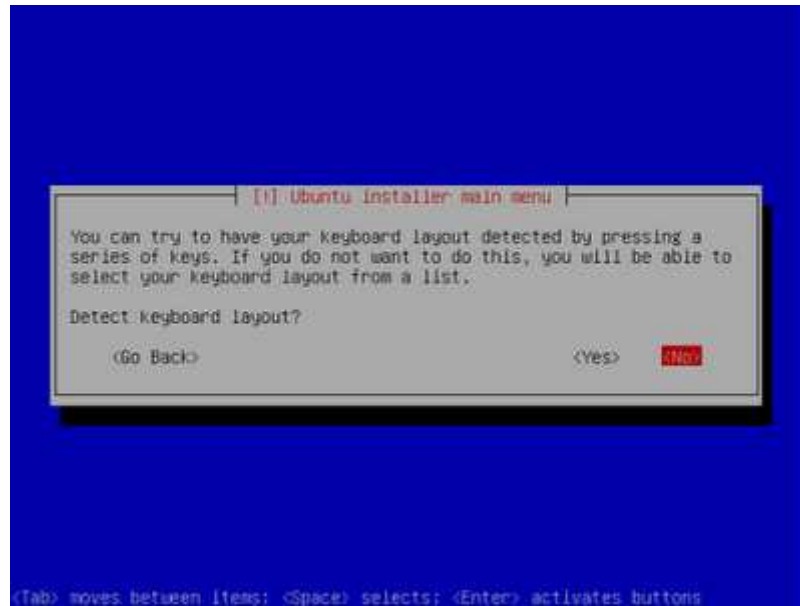


5. เลื่อนลงมาจนเจอ Thailand



6. ระบบติดตั้งจะตรวจสอบ keyboard layout ถ้าต้องการ ตอบ Yes ถ้าไม่ต้องการตอบ No ในที่นี้

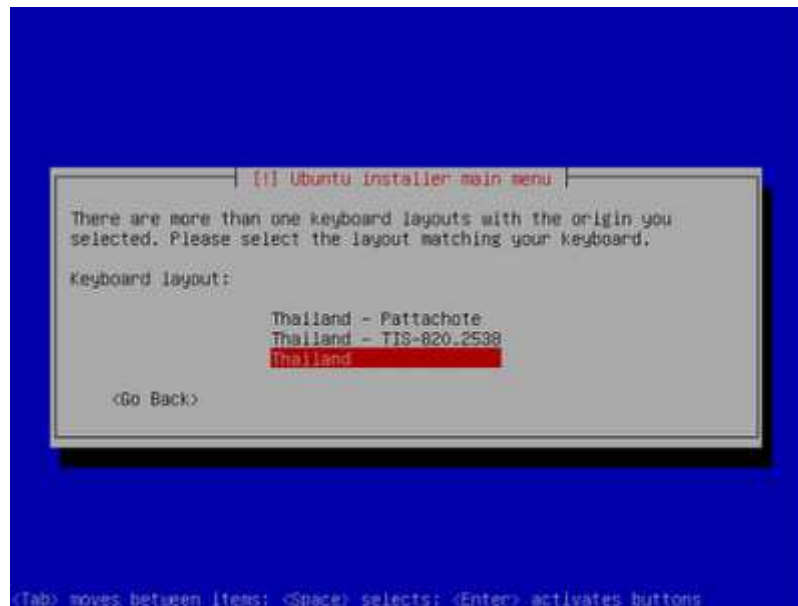
ตอบ No



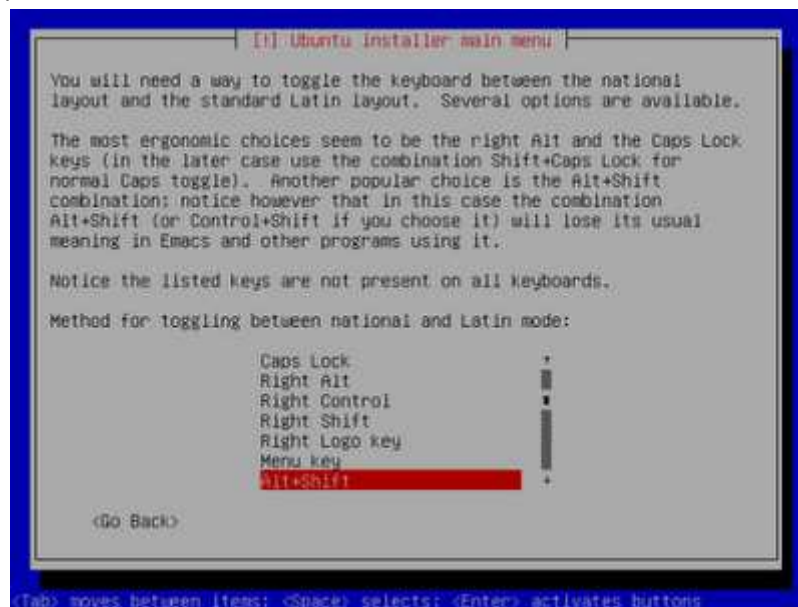
7. เลือกคีย์บอร์ดเป็น Thailand



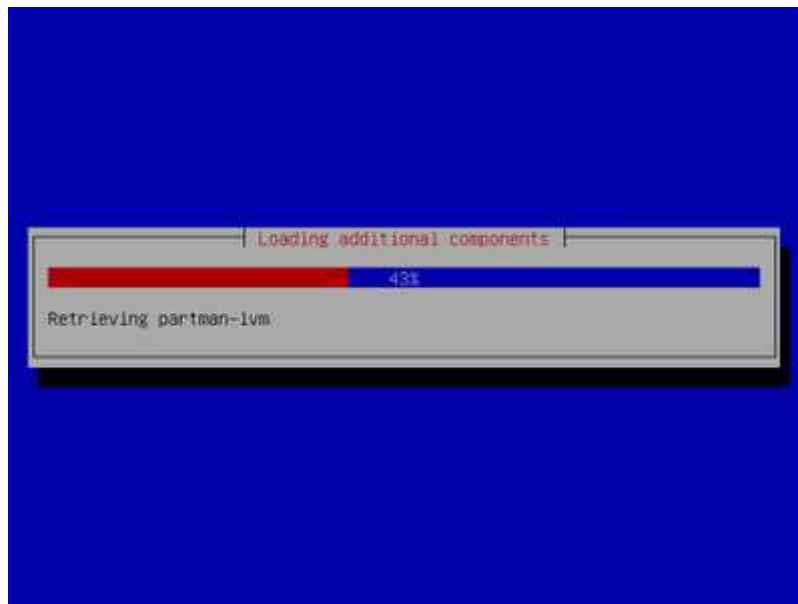
8. เลือก keyboard layout เป็น Thailand



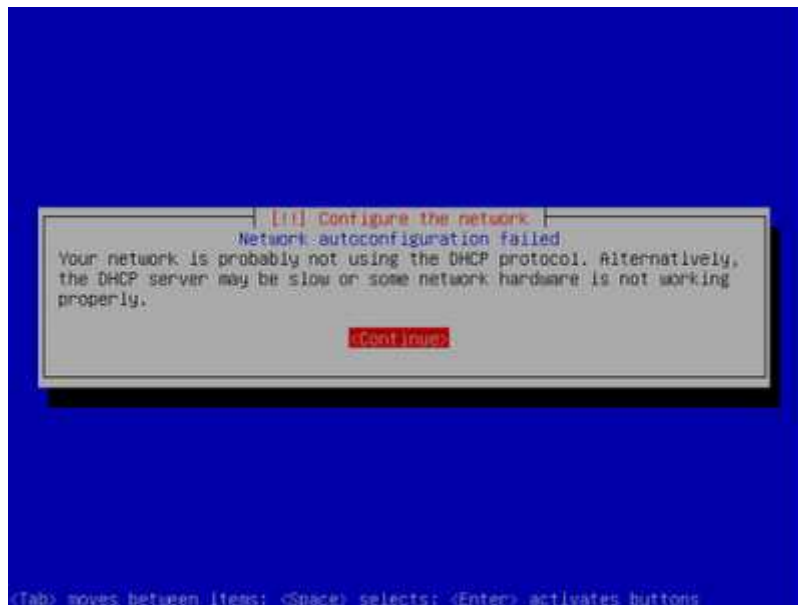
9. เลือกปุ่มสลับภาษา Alt+Shift



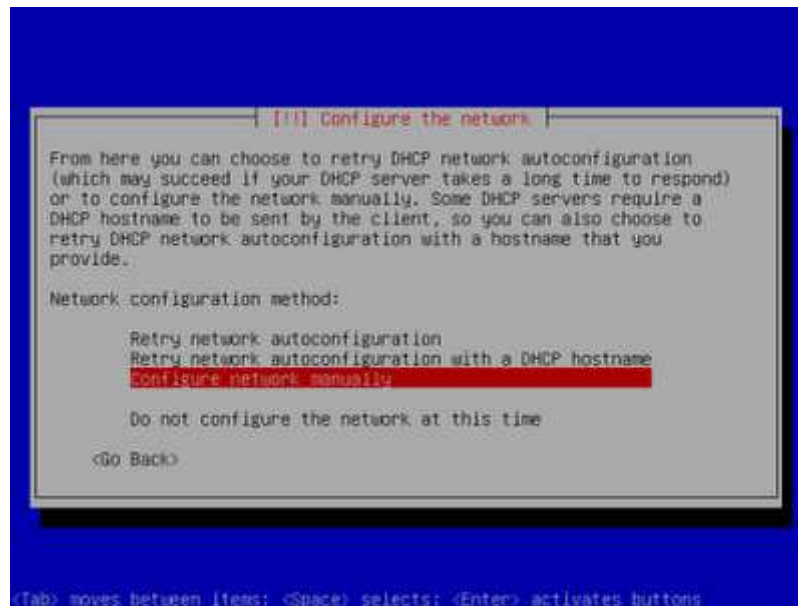
10. ระบบติดตั้งจะหา DHCP Server



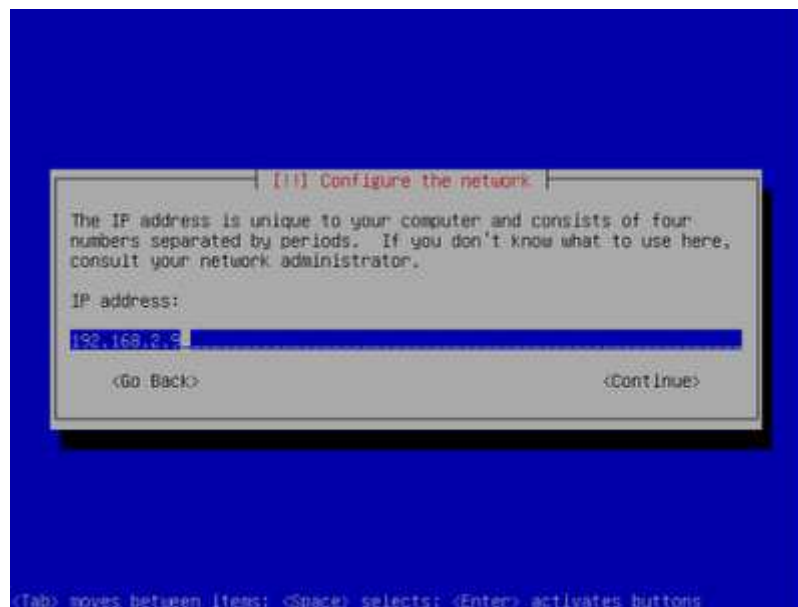
11. ถ้าไม่มี DHCP Server อยู่ในเน็ตเวิร์ค ระบบติดตั้งก็จะรายงานว่าหาไม่เจอ



12. ต้องคอนฟิกเน็ตเวิร์คเองโดยเลือก Configure network manual



13. ใส่ค่า IP address ที่ต้องการ



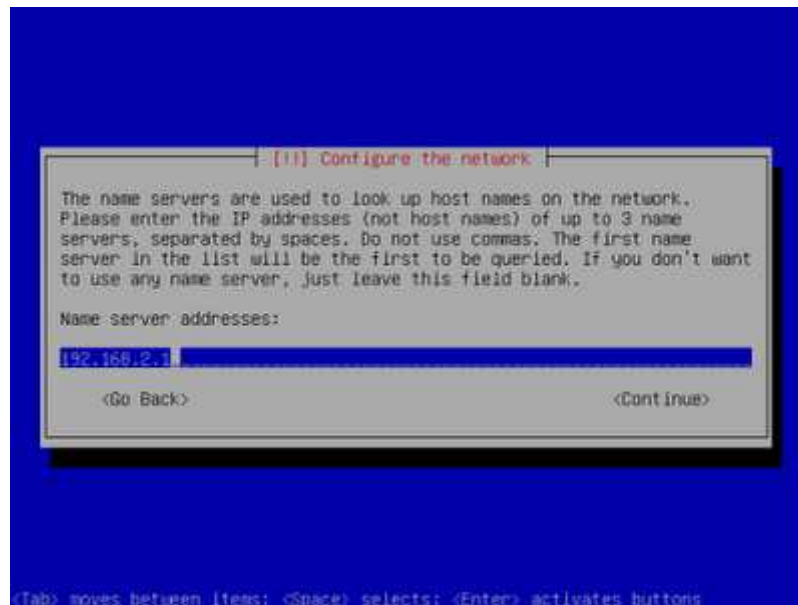
ใส่ค่า Netmask



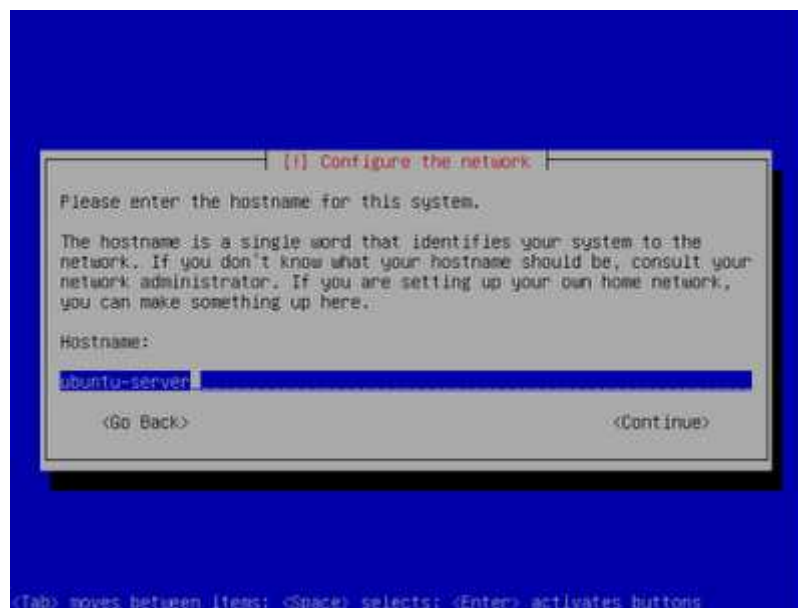
14. ใส่ค่า Gateway ถ้าไม่ต้องการให้เครื่องที่เราติดตั้งใช้งานอินเทอร์เน็ต ก็ปล่อยว่างไว้ได้



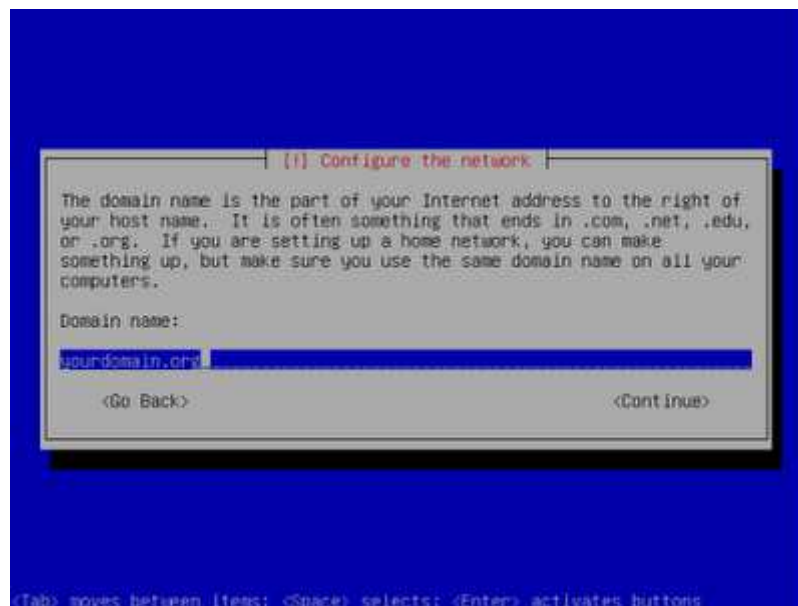
15. ใส่ค่า Name server address ถ้าไม่มี Name server ภายในก็สามารถใส่ ค่า Name server ของ ISP ที่เราใช้บริการอยู่ เช่น KSC 203.155.33.1 ถ้าไม่ต้องการให้เครื่องที่เราติดตั้งใช้งานอินเทอร์เน็ตก็เว้นว่างไว้ได้



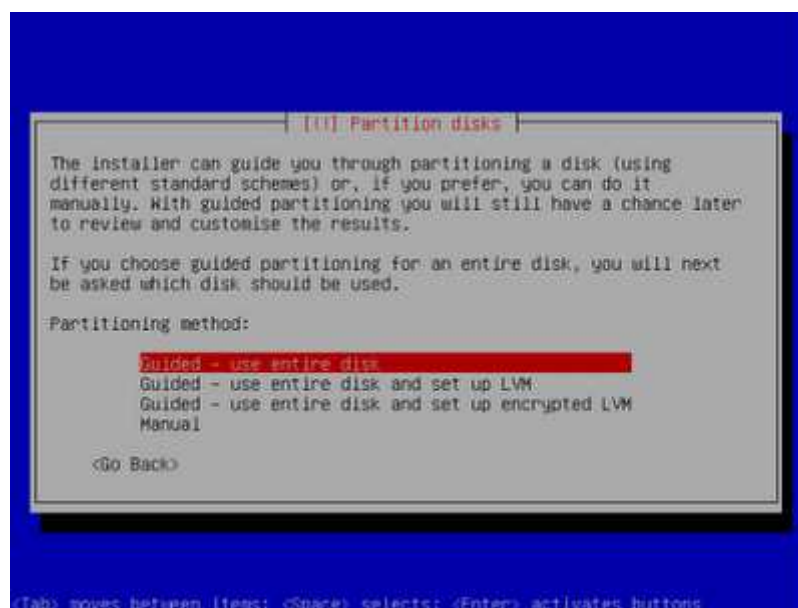
16. ตั้งชื่อเครื่องชื่ออะไรก็ได้ตามที่เรต้องการ



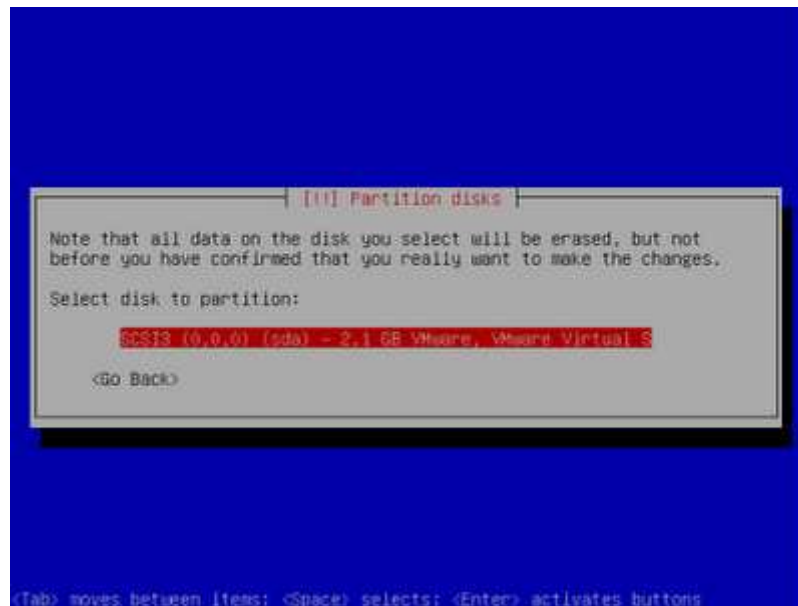
17. ตั้งชื่อโดเมน



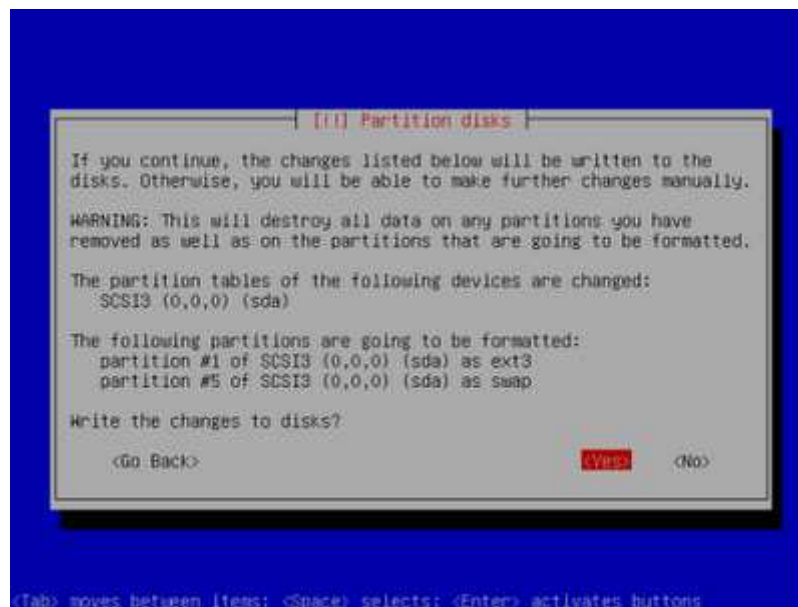
18. พาร์ติชันฮาร์ดดิสก์ ก็มีวิธีการต่างๆ เพื่อความสะดวกเลือก Guide – use entire disk โดยที่
 ต้องไม่มีข้อมูลที่ต้องการอยู่ในฮาร์ดดิสก์ ขั้นตอนการพาร์ติชันที่แนะนำตรงนี้ เพื่อให้สะดวกสำหรับ
 มือใหม่ ในการใช้งานจริงจะแนะนำการพาร์ติชันอีกทีหนึ่ง



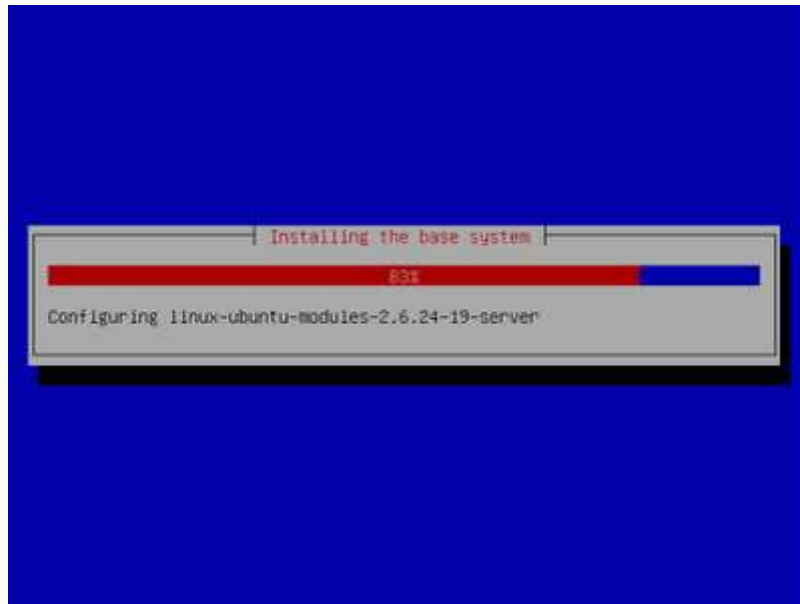
19. เลือกฮาร์ดดิสก์ที่ต้องการพาร์ติชัน



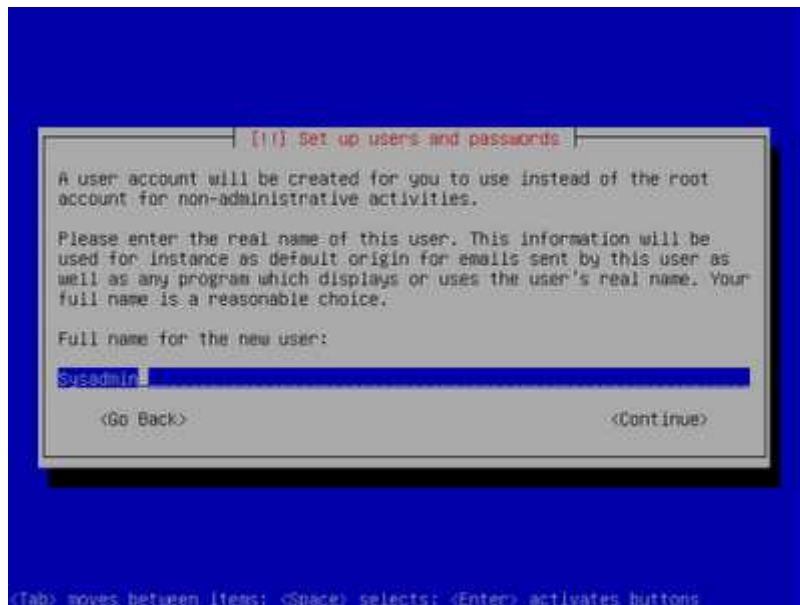
20. ระบบติดตั้งก็จะคำนวณขนาดพาร์ติชันให้เราเรียบร้อยแล้ว



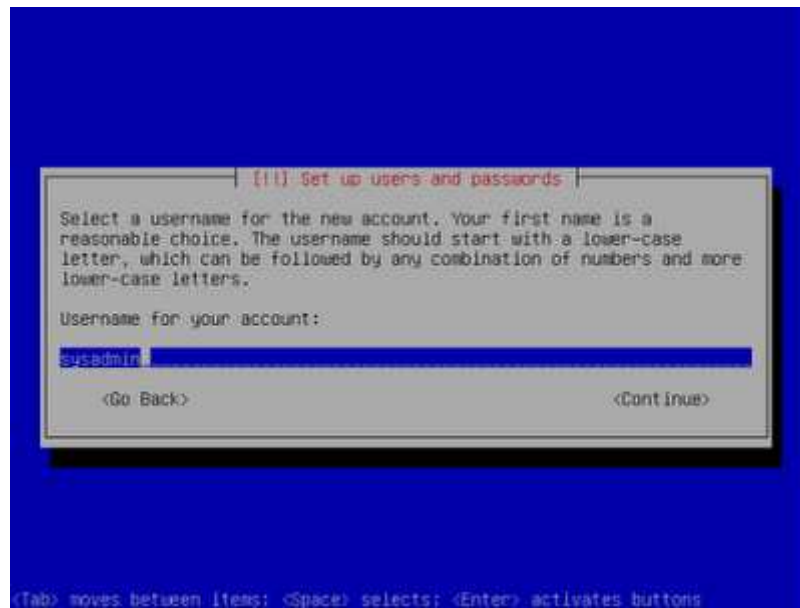
21. เมื่อทำการพาร์ติชันเสร็จก็จะทำการติดตั้งระบบหลักของ Ubuntu



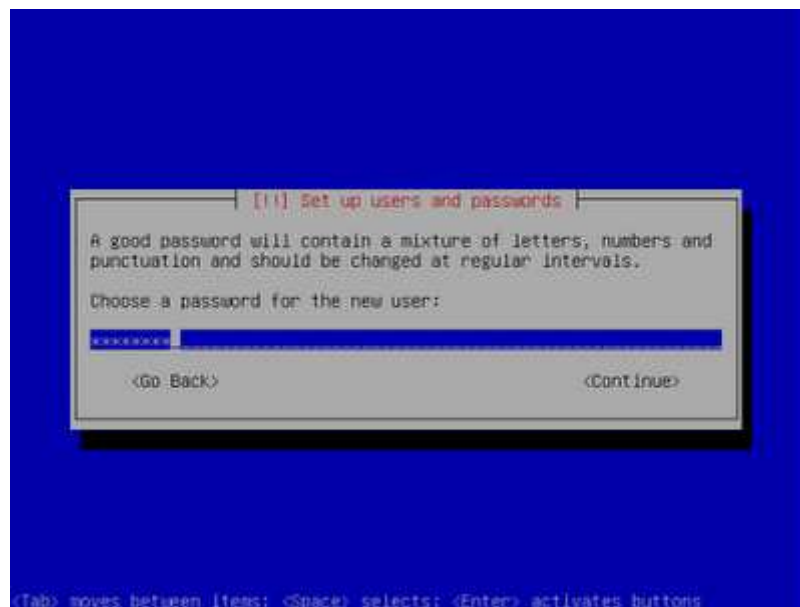
22. ระบุชื่อผู้ใช้งานที่ใช้สำหรับ login สำหรับ Ubuntu จะไม่มีการระบุผู้ใช้ root



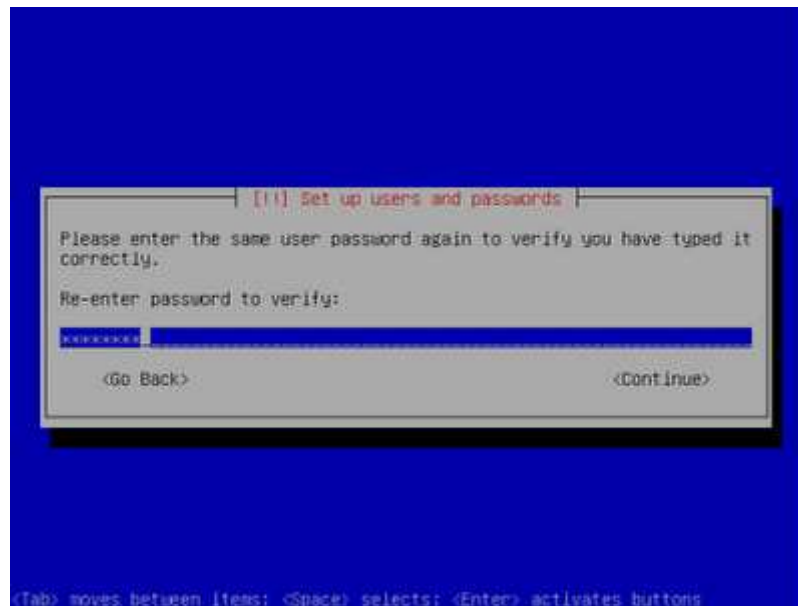
23. ใส่ username ที่จะใช้ Login



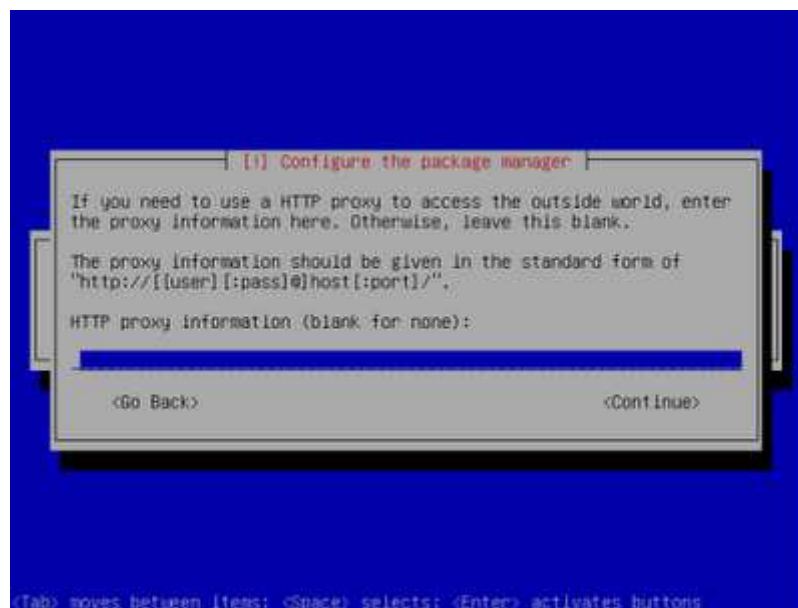
24. กำหนดรหัสผ่านของ user



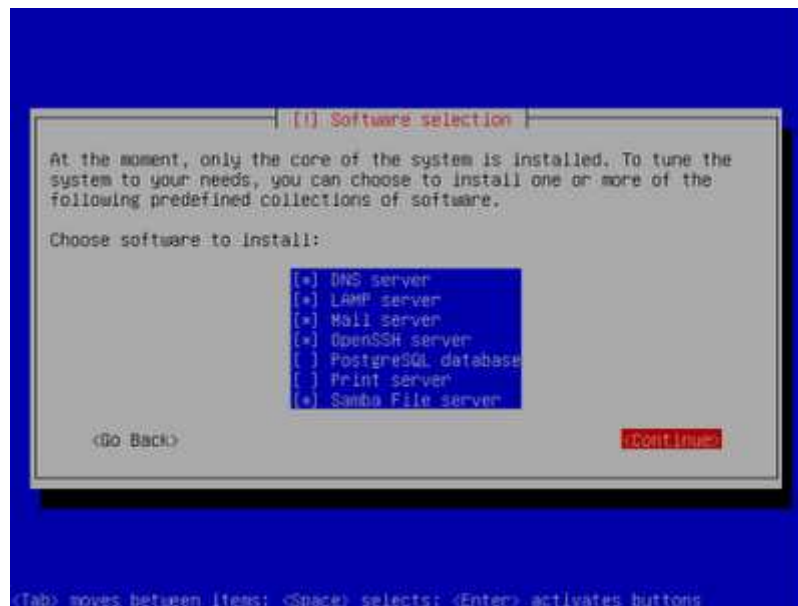
กำหนดรหัสผ่านอีกครั้ง โดยใส่ให้เหมือนกับครั้งแรก



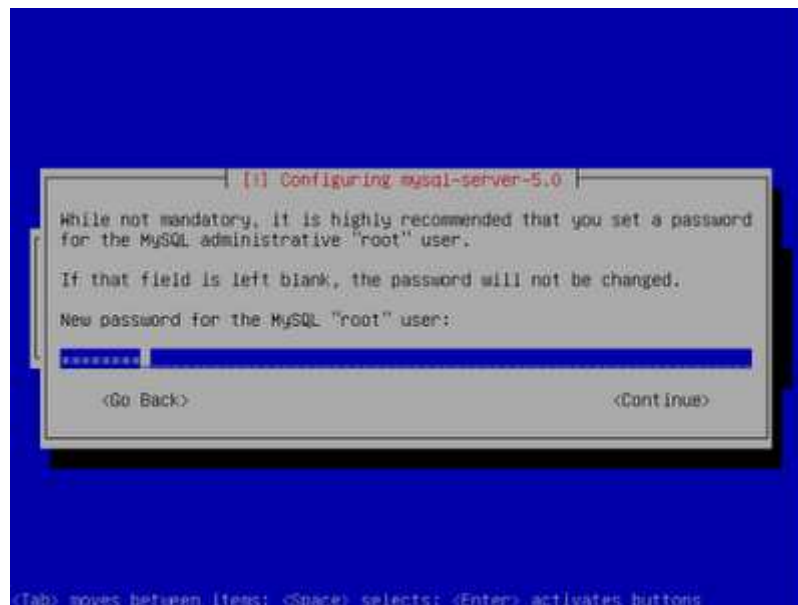
25. ถ้าในเน็ตเวิร์คอยู่หลัง proxy และเป็น proxy authentication ก็ให้ระบุ รายละเอียดของ proxy ลงไป เช่น <http://lampangnet:password@192.168.2.254:8080> ถ้าไม่มีก็เว้นว่างเอาไว้



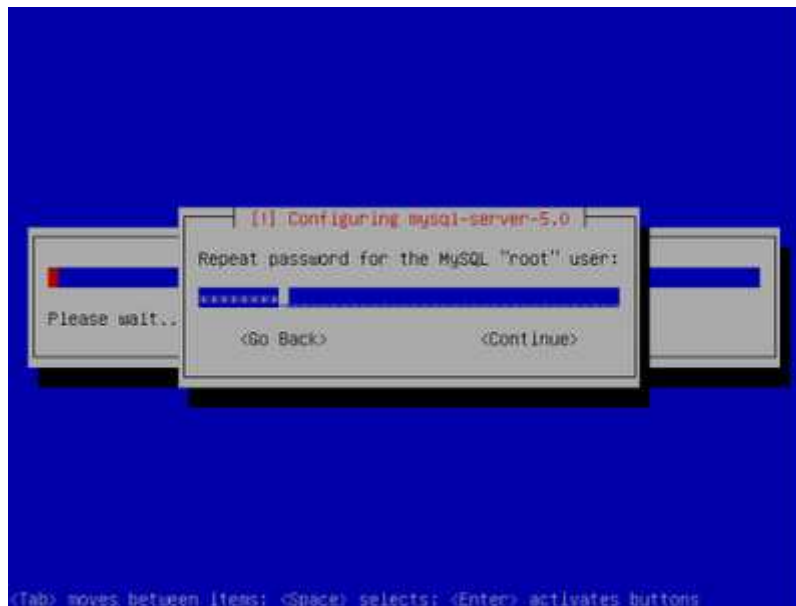
26. เลือกโปรแกรมที่จะติดตั้ง สามารถเลือกใหม่ได้ในภายหลังด้วยคำสั่ง **tasksel**



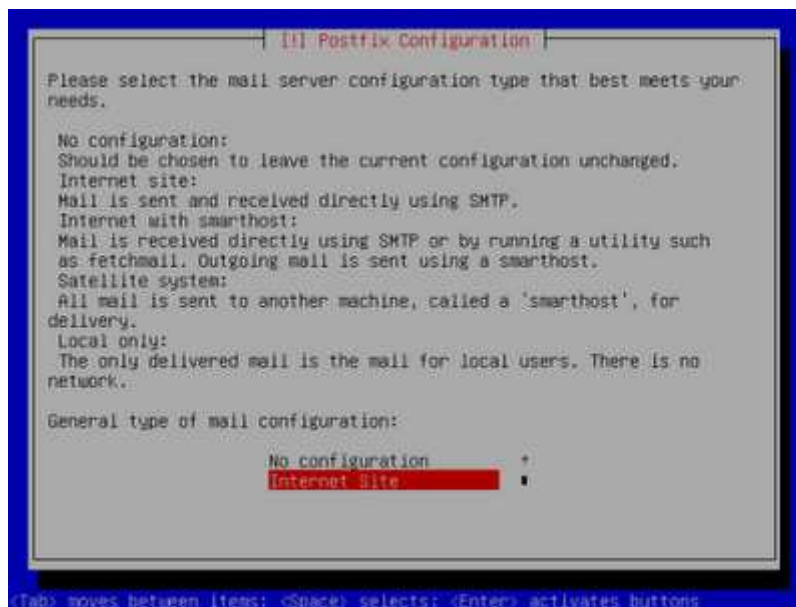
27. ถ้าเลือก LAMP server ก็จะทำให้ระบุรหัสผ่านของ user root ของ MySQL

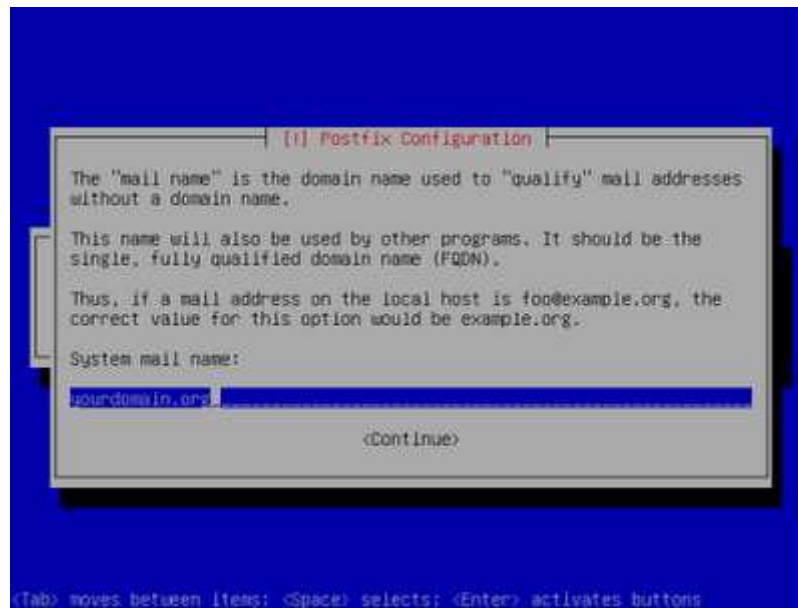


ใส่รหัสผ่านของ MySQL อีกครั้ง

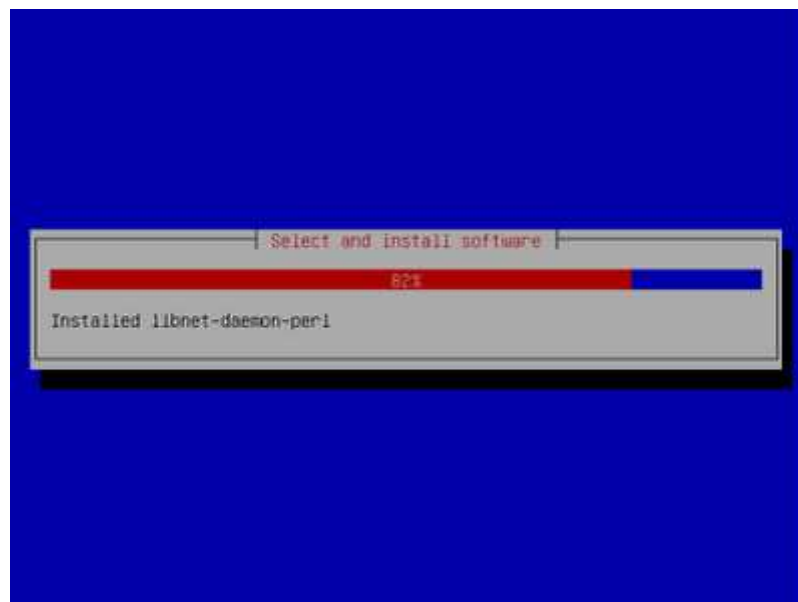


28. ถ้าเลือก Mail server ก็จะมีขั้นตอนให้คอนฟิก Postfix

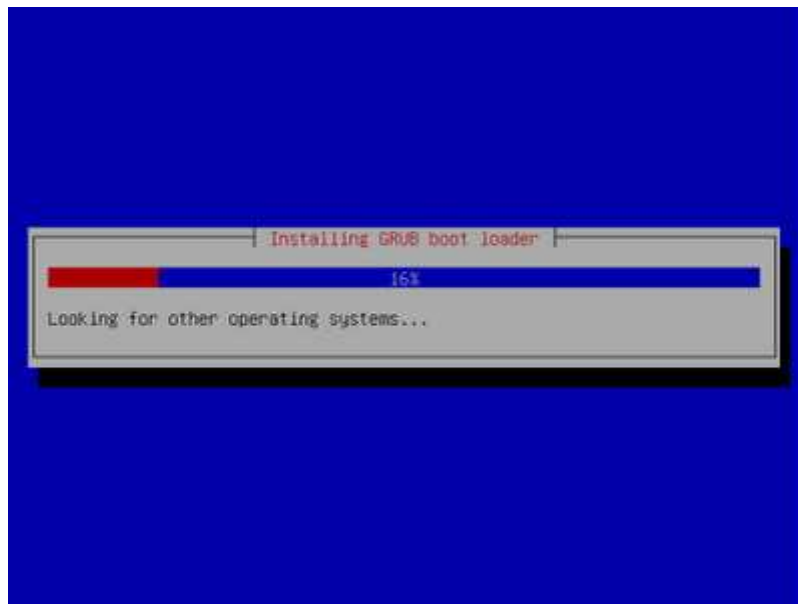




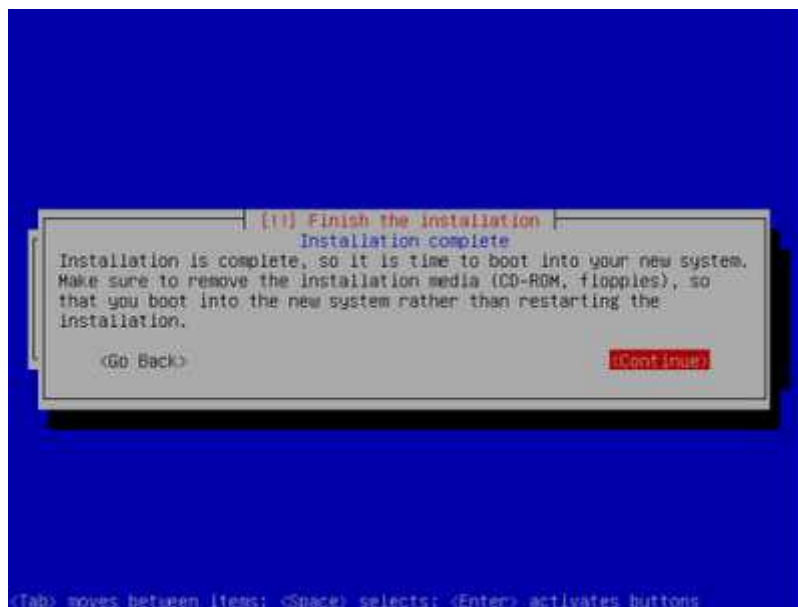
29. ระบบทำการติดตั้งโปรแกรมที่เราได้เลือกไป



30. ระบบทำการติดตั้ง GRUB Boot loader



31. การติดตั้ง Ubuntu Server เสร็จเรียบร้อย



(อ้างอิงจาก <http://www.sothorn.org>)

คำสั่งที่จำเป็นใช้งานใน *Ubuntu*

ทำไมต้อง *sudo*

ปกติผู้เป็นเจ้าของหรือเจ้าของเครื่อง ผู้ดูแลระบบ จะเคยชินกับการล็อกอินด้วยผู้ใช้ *root* และใช้คำสั่งด้วยผู้ใช้ *root* ก็ไม่เป็นไรถ้าผู้ดูแลระบบมีคนเดียว ถ้าผู้ดูแลระบบในเครื่องนั้นๆ มีหลายคน เมื่อมีปัญหาขึ้นมาก็จะตามหาว่าใครเป็นคนแก้ไข อะไร ตรงไหนบ้าง

sudo เป็นเครื่องมือที่อนุญาตให้ผู้ใช้บางคน หรือบางกลุ่ม สามารถมีสิทธิ์ในการใช้คำสั่งได้เหมือนผู้ใช้ *root* หรือในบางที่สามารถกำหนดให้มีสิทธิ์เท่าเทียมกับ *root* ก็ได้ ซึ่งขึ้นอยู่กับ การปรับแต่งค่าต่างๆ ในไฟล์ `/etc/sudoers`

ในขั้นตอนของการติดตั้ง *ubuntu* จะเห็นว่าไม่มีให้เราใส่รหัสผ่านของผู้ใช้ *root* แต่จะมีขั้นตอนของการเพิ่มผู้ใช้ และรหัสผ่านอื่นอีกหนึ่งคน เมื่อเราติดตั้งเสร็จเราก็ไม่สามารถเข้าระบบได้ด้วยผู้ใช้ *root* การที่เราเข้าระบบด้วยผู้ใช้ทั่วไปเราก็ไม่สามารถจะใช้คำสั่งที่เป็นสิทธิ ของผู้ใช้ *root* ได้ จะใช้ได้ก็ต่อเมื่อเราใช้คำสั่ง ***sudo*** ร่วมด้วยเช่น

```
sysadmin@ubuntu-server:~$ apt-get update
E: Could not open lock file /var/lib/apt/lists/lock - open (13 Permission denied)
E: Unable to lock the list directory
```

จะเห็นว่าไม่สามารถเรียกใช้คำสั่งได้

ถ้าใช้ *sudo* ก็จะเรียกใช้คำสั่งได้ ดังตัวอย่างนี้

```
sysadmin@ubuntu-server:~$ sudo apt-get update
[sudo] password for sysadmin:
0% [Connecting to th.archive.ubuntu.com]
```

คำสั่งพื้นฐานการใช้งานระบบปฏิบัติการ UNIX (Linux)

คำสั่งเกี่ยวกับ *File System*

Path คือที่อยู่ของ File หรือ Directory ในระบบ Unix แบ่งเป็น 2 ชนิดคือ Absolute Path อ้างอิงจาก / (เรียกว่า root directory เป็น Directory เริ่มต้นของระบบ File) ตัวอย่าง /etc/passwd เป็นที่อยู่ของ File ที่เก็บรายละเอียดของผู้ใช้งานในระบบ

Relative Path อ้างอิงจาก Directory ที่ทำงานอยู่ปัจจุบัน (Working Directory) การอ้างอิงแบบ Relative มีสัญลักษณ์แทน Directory ปัจจุบันเป็น . (จุด) และ Directory ที่อยู่เหนือขึ้นไปหนึ่งระดับ (Parent Directory) แทนด้วย .. (จุดสองจุดติดกัน)

Wildcard เป็นสัญลักษณ์แทนชื่อ File หรือ Directory

* แทนตัวอักษรที่ตัวก็ได้ (ชื่อ File เป็นอะไรก็ได้)

? แทน 1 ตัวอักษร

Working Directory การอ้างอิงถึง File หรือ Directory ที่ไม่เป็นแบบ Absolute Path จะเป็นการอ้างอิงกับ Working Directory เสมอ การแสดง Working Directory หลังจาก login เข้าสู่ระบบแล้ว จะแสดง Command Prompt ของ Shell ถ้าต้องการแสดง Directory ปัจจุบันที่ทำงานอยู่ใช้คำสั่ง **pwd** (Print Working Directory) ได้ผลลัพธ์ดังนี้

```
#pwd
/root
#
```

แสดงว่าปัจจุบันทำงานอยู่ที่ Directory /root (Working Directory)

การเปลี่ยน Working Directory

ใช้คำสั่ง **cd** (Change Directory) แล้วตามด้วยที่อยู่ของ Directory เช่น ถ้าต้องการเปลี่ยนไปทำงานที่ Directory /etc ใช้คำสั่ง **cd /etc** แล้วตามด้วยคำสั่ง **pwd** เพื่อแสดง Directory ปัจจุบันที่เปลี่ยนไป

```
#cd /etc/rc.d
#pwd
/etc/rc.d
```

การแสดงรายชื่อ File และ Directory ย่อย

ใช้คำสั่ง `ls` (List) แล้วตามด้วยที่ Path ของ Directory ที่ต้องการแสดงรายละเอียด ถ้าไม่ใส่จะหมายถึงแสดงรายละเอียดของ Directory ปัจจุบันที่ทำงานอยู่

```
#ls
init.d/ rc* rc0.d/ rc1.d/ rc2.d/ rc3.d/ rc4.d/ rc5.d/
rc6.d/ rc.local* rc.modules* rc.sysinit*
```

Option ที่ใช้งาน `-l` ใช้แสดงรายละเอียดทั้งหมด `-a` ใช้แสดง hidden File หรือ Directory โดย Hidden File และ Hidden Directory ใน Unix คือ File หรือ Directory ที่มีชื่อขึ้นต้นด้วย `.` (จุด)

การจัดการ File

การสร้าง File สามารถทำได้หลายวิธี ถ้าต้องการ File ขนาด 0 byte สามารถใช้คำสั่ง `touch` แล้วตามด้วยชื่อ File

```
#touch unix.txt
#ls -l
total 0
-rw-r--r-- 1 root root 0 Jan 15 07:19 unix.txt
```

การ copy File ใช้คำสั่ง `cp <ชื่อ File ต้นฉบับ> <ชื่อ File สำเนา>`

```
#cp unix.txt data.txt
#ls -l
total 0
-rw-r--r-- 1 root root 0 Jan 15 07:20 data.txt
-rw-r--r-- 1 root root 0 Jan 15 07:19 unix.txt
```

การย้าย File (เปลี่ยนชื่อ File คือการย้าย File ไว้ใน Directory เดิมแต่ใช้ชื่อต่างจากชื่อเดิม) ใช้คำสั่ง `mv <ชื่อ File เดิม> <ชื่อ File ใหม่>`

```
#mv unix.txt linux.txt

#ls -l

total 0

-rw-r--r-- 1 root root 0 Jan 15 07:20 data.txt
-rw-r--r-- 1 root root 0 Jan 15 07:19 linux.txt
```

การลบ File ใช้คำสั่ง `rm` <ชื่อ File ที่ต้องการลบ>

```
#rm linux.txt

rm: remove regular empty file `linux.txt'? y

#ls -l

-rw-r--r-- 1 root root 0 Jan 15 07:20 data.txt
```

การจัดการเกี่ยวกับ Directory

สร้าง Directory ใช้คำสั่ง `mkdir` <ชื่อ Directory >

```
#mkdir web

#ls -l

total 4

drwxr-xr-x 2 root root 4096 Jan 15 07:17 web/
```

การสร้าง Directory แบบหลายๆ ชั้น ใส่ *Option -p* (ตามปกติเราสามารถสร้าง Directory ได้ครั้งละชั้น)

```
#mkdir -p /user/web/data

ย้าย Directory (เปลี่ยนชื่อ) mv <Source Path> <Destination Path>

#mv web ftp

#ls -l

total 4

drwxr-xr-x 2 root root 4096 Jan 15 07:17 ftp/
```

ลบ Directory ใช้คำสั่ง `rmdir <ชื่อ Directory >`

```
#rmdir ftp
#ls -l
total 0
```

คำสั่งแสดงข้อมูลใน File

คำสั่ง `cat <ตามด้วยชื่อ File>`

```
#cat /etc/passwd
```

เป็นการแสดงรายละเอียดของผู้ใช้งานในระบบ การใช้คำสั่ง `cat` จะแสดงข้อมูลทั้งหมดออกที่หน้าจอถ้าข้อมูลใน File มีเกินกว่าจำนวนบรรทัดบนหน้าจอเราจะเห็นเฉพาะหน้าจอสุดท้าย

```
#cat /var/log/messages
```

คำสั่ง `more <ตามด้วยชื่อ File>` เป็นการแสดงข้อมูลใน File ออกทางหน้าจอครั้งละหน้าจอ กด Spacebar ถ้าต้องการแสดงข้อมูลหน้าจอถัดไป หรือกด Enter เพื่อแสดงข้อมูลบรรทัดต่อไป กด q เพื่อออกจากคำสั่งก่อนการแสดงผลจะจบ ระหว่างการแสดงถ้าต้องการค้นหาข้อความใดสามารถ พิมพ์ / แล้วตามด้วยข้อความนั้นแล้วกด Enter

```
#more /var/log/messages
```

(คำสั่ง `less` ทำงานได้เช่นเดียวกับคำสั่ง `more` แต่มีรายละเอียดมากกว่า)

```
#less /var/log/messages
```

คำสั่งที่ใช้นำจำนวนตัวอักษร จำนวนบรรทัด จำนวนคำใน File

`wc <ตามด้วยชื่อ File`

```
#wc /etc/passwd
39 91 1886 /etc/passwd
```

จากผลลัพธ์ File /etc/passwd มี 39 บรรทัด 91 คำ และ 1886 ตัวอักษรตามลำดับ

คำสั่งกรองข้อมูล

grep <ข้อความที่ต้องการ> <ชื่อ File>

ต้องการหาคำว่า root จาก File /etc/passwd คำสั่งที่เรียกใช้คือ

```
#grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

ผลลัพธ์ที่ได้จะแสดงเฉพาะบรรทัดที่อยู่ใน File /etc/passwd ซึ่งมีคำว่า root ต้องการหาคำว่า

DocumentRoot จาก File /etc/httpd/conf/httpd.conf

```
#grep DocumentRoot /etc/httpd/conf/httpd.conf
DocumentRoot /var/www/html
```

แสดงผลลัพธ์จากคำสั่ง sysctl เฉพาะบรรทัดที่มีคำว่า forward

```
[root@garfield root]# sysctl -a |grep forward
net.ipv4.conf.eth1.mc_forwarding = 0
net.ipv4.conf.eth1.forwarding = 1
net.ipv4.conf.default.mc_forwarding = 0
net.ipv4.conf.default.forwarding = 1
net.ipv4.conf.all.mc_forwarding = 0
net.ipv4.conf.all.forwarding = 1
net.ipv4.ip_forward = 1
```

คำสั่งใช้สำหรับค้นหา File

คำสั่ง `sort <ชื่อไฟล์>`

```
#sort /etc/passwd
adm:x:3:4:adm:/var/adm:/bin/sh
alias:x:81:11:alias user:/var/qmail/alias:/bin/true
amavis:x:97:502:Anti Virus Checker:/var/virusmails:/bin/false
apache:x:72:72:system user for apache:/var/www:/bin/sh
bin:x:1:1:bin:/bin:/bin/sh
clamav:x:90:90:system user for clamav:/var/lib/clamav:/bin/sh
daemon:x:2:2:daemon:/sbin:/bin/sh
ftp:x:76:76:system user for proftpd:/var/ftp:/bin/false
games:x:12:100:games:/usr/games:/bin/sh
```

คำสั่ง `find` ใช้สำหรับค้นหา File

คำสั่ง `find <start Directory> -name <filename>`

ค้นหา file ที่ชื่อ `wget` เริ่มต้นจาก root Directory /

```
#find / -name wget
find: /mnt/floppy: Input/output error
find: /mnt/cdrom: Input/output error
/usr/bin/wget
```

`filename` สามารถกำหนดโดยใช้ wildcard (* ?) แทนได้แต่ให้ครอบไว้ด้วย ‘ (single quote)

ค้นหา file ที่มีชื่อขึ้นต้นด้วย `syslog` แล้วตามด้วยอะไรก็ได้ เริ่มต้นจาก root Directory /

```
#find / -name 'syslog*'
/etc/sysconfig/syslog
/etc/rc.d/init.d/syslog
/etc/logrotate.d/syslog
/etc/syslog.conf
/etc/webmin/syslog
```

** คู่มือละเอียดเพิ่มเติมจากคำสั่ง **man find**

คำสั่งสำหรับสร้าง Link

Link แบ่งเป็น 2 ชนิดคือ

- **hard link** ไม่สามารถ link ข้าม file system เป็นการชี้ไปที่ data block ของ file นั้นๆ ถ้าลบ file ต้นฉบับยังสามารถเข้าถึงข้อมูลของ file นั้นผ่าน hard link ได้
- **symbolic link** สามารถ link ข้าม file system ได้ เป็นการชี้ไปยังชื่อ file ต้นฉบับ ถ้าลบ file ต้นฉบับ จะไม่สามารถเข้าถึงข้อมูลใน file นั้นได้

คำสั่ง **ln <Source> <Link>**

```
# cp /etc/hosts ./
# ls -l
total 4
-rw-r--r-- 1 root root 93 Jan 15 09:34 hosts
# ln hosts myhosts
# ls -l
total 8
-rw-r--r-- 2 root root 93 Jan 15 09:34 hosts
-rw-r--r-- 2 root root 93 Jan 15 09:34 myhosts
# cat hosts
195.168.1.254 garfield.info.com
# cat myhosts
195.168.1.254 garfield.info.com
# rm hosts
rm: remove regular file `hosts'? y
# ls -l
total 4
-rw-r--r-- 1 root root 93 Jan 15 09:34 myhosts
# cat myhosts
195.168.1.254 garfield.info.com
```

ถ้าต้องการสร้าง **symbolic link** ให้ใช้ *option -s*

```
# ls -l /var/log/messages
-rw-r----- 1 root adm 638744 Jan 15 09:23 /var/log/messages
# ln -s /var/log/messages systemlog
# ls -l
total 0
lrwxrwxrwx 1 root root 17 Jan 15 09:23 systemlog ->
/var/log/messages
# more systemlog
```

คำสั่ง ใช้สำหรับ Format File System

คำสั่ง mkfs

```
# mkfs /dev/sda8
mke2fs 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
28112 inodes, 112423 blocks
5621 blocks (5.00%) reserved for the super user
First data block=1
14 block groups
8192 blocks per group, 8192 fragments per group
2008 inodes per group
Superblock backups stored on blocks:
8193, 24577, 40961, 57345, 73729
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

File Compression

คำสั่ง **tar**

```
tar cvf <file.tar> <source path>
```

```
# mkdir backup
# cd backup
# tar cvf etc.tar /etc/
# ls -l
total 9048
-rw-r--r-- 1 root root 9246720 Jan 14 12:57 etc.tar
```

คำสั่ง **gzip/gunzip**

```
gzip <filename>
```

```
gunzip <filename.gz>
```

```
# gzip etc.tar
# ls -l
total 1636
-rw-r--r-- 1 root root 1668422 Jan 14 12:57 etc.tar.gz
# ls -l
total 1636
-rw-r--r-- 1 root root 1668422 Jan 14 12:57 etc.tar.gz
# gunzip etc.tar.gz
# ls -l
total 9048
-rw-r--r-- 1 root root 9246720 Jan 14 12:57 etc.tar
# tar xvf etc.tar
# ls -l
total 9052
drwxr-xr-x 70 root root 4096 Jan 14 12:40 etc/
-rw-r--r-- 1 root root 9246720 Jan 14 12:57 etc.tar
```

คำสั่ง zip/unzip

zip <filename>

unzip <filename.zip>

```
$ zip etc.zip /etc/*
$ ls -l
total 460
-rw-r--r-- 1 mong users 465312 Jan 14 13:51 etc.zip
$ ls -l
total 460
-rw-r--r-- 1 mong users 465312 Jan 14 13:51 etc.zip
$ unzip etc.zip
$ ls -l
total 464
drwxr-xr-x 48 mong users 4096 Dec 28 17:31 etc
-rw-r--r-- 1 mong users 465312 Jan 14 13:51 etc.zip
```

คำสั่งเกี่ยวกับ Process

Process คือ โปรแกรมที่กำลังทำงานอยู่ (มีการใช้งาน Memory และ CPU) โปรแกรมที่เก็บไว้ใน Hard Disk เป็นเพียงชุดคำสั่งที่ยังไม่ได้ทำงาน แต่ถ้าถูกเรียกใช้งานจะมีการจอง Resource (Memory CPU) แล้วนำโปรแกรมไปทำงาน ดังนั้นโปรแกรมเดียวกันสามารถถูกเรียกใช้งานเป็น Process ได้หลายๆ

Process

Process แต่ละ Process มีหมายเลข Process เรียกว่า Process ID (PID) ซึ่งเป็นตัวเลขที่ไม่ซ้ำกัน เพื่อใช้อ้างอิงและควบคุมการทำงานของ Process ต่างๆ โดยผู้ที่สามารถควบคุม Process ได้ต้องเป็นผู้ที่สร้าง Process นั้นขึ้นมา (เป็นเจ้าของ Process) โดยมีข้อยกเว้นว่า root สามารถควบคุม Process ทุกๆ Process ในระบบ

คำสั่งที่ใช้แสดงรายละเอียดเกี่ยวกับ Process

ใช้คำสั่ง ps

```
# ps
PID TTY TIME CMD
23633 pts/1 00:00:00 bash
27171 pts/1 00:00:00 ps
```

```
# ps -ef
UID PID PPID C STIME TTY TIME CMD
root 1 0 0 Dec17 ? 00:00:04 init [3]
root 2 1 0 Dec17 ? 00:00:00 [keventd]
root 3 1 0 Dec17 ? 00:00:00 [kapmd]
root 4 1 0 Dec17 ? 00:00:00 [ksoftirqd_CPU0]
root 5 1 0 Dec17 ? 00:00:06 [kswapd]
.... ผลลัพธ์ที่เหลือถูกกลบไว้ ....
```

(ถ้าต้องการแสดงผลครั้งละหน้าจอลำดับที่เรียกใช้จะเป็น ps -ef | more)

Option e กำหนดให้แสดง Process ทั้งหมดในระบบ Option f แสดงรายละเอียดทั้งหมด จากผลลัพธ์แต่ละ Column มีรายละเอียดดังนี้

UID แสดงเจ้าของ Process

PID แสดง Process ID

PPID แสดง Parent Process ID (Process ที่เป็นคนสร้าง Process นี้ขึ้นมา)

CMD คำสั่งที่ถูกเรียกให้ทำงานเป็น Process

การใช้งาน vi เบื้องต้น

สร้าง File ใหม่โดยใช้คำสั่ง vi <File Name> ถ้ามี File อยู่แล้วจะเป็นการเปิด File เพื่อแก้ไข

```
#vi file1.txt
```

จะเข้าสู่ Command Mode ถ้าต้องการเพิ่มข้อมูลให้พิมพ์คำสั่ง i (insert) หรือ คำสั่ง a (append) หลังจากพิมพ์คำสั่ง i หรือ a จะไม่แสดงตัว i หรือ a ขึ้นมาแต่จะเข้าสู่ Edit Mode ที่สามารถเพิ่มข้อมูลลง

ไปใน File ได้ ถ้าต้องการบันทึกข้อมูลให้กลับสู่ Command Mode โดยการกดปุ่ม Esc แล้วทำการบันทึกด้วยการพิมพ์คำสั่ง :w แล้ว Enter และออกจาก vi ด้วยคำสั่ง :q

การค้นหาข้อความใน Editor vi ใน Command Mode ใช้คำสั่ง / แล้วตามด้วยข้อความที่ต้องการค้นหา

```
#vi /etc/httpd/conf/httpd.conf
```

ถ้าต้องการหาคำว่า DocumentRoot ใช้คำสั่ง /DocumentRoot แล้ว Enter จากนั้น Cursor จะไปอยู่ที่ตำแหน่งของคำแรกที่ค้นพบ ถ้าต้องการแสดงผลลัพธ์ต่อไปให้พิมพ์คำสั่ง / แล้ว Enter การค้นหาด้วยคำสั่ง / เป็นการค้นหาจากด้านบนของ File ไปด้านล่าง ถ้าต้องการค้นหาย้อนกลับให้เปลี่ยนจาก / เป็น ? การไปยังบรรทัดที่ต้องการให้พิมพ์ตัวเลขของบรรทัดแล้วตามด้วย G (จัตวาใหญ่) จากนั้น Cursor จะไปยังบรรทัดที่ต้องการ ถ้าต้องการไปที่บรรทัดสุดท้ายของ File ให้พิมพ์ G (ไม่ต้องใส่หมายเลขบรรทัด)

คำสั่งเพื่อใช้งาน File System

การนำ File System มาใช้งานต้องทำการ mount คือการกำหนดรายละเอียดของ File System และจุดต่อเชื่อม (Mount Point) Mount Point คือ Directory ที่ใช้บนจุดต่อเชื่อมซึ่งอยู่บน File System หลัก ภายใน Directory นี้ไม่ควรข้อมูลเก็บอยู่ เพราะขณะที่ทำการ mount แล้วจะไม่สามารถเข้าใช้งานข้อมูลที่อยู่ใน Directory ได้ เนื่องจากภายใต้ Directory เป็นข้อมูลของ File System ที่ทำการ mount อยู่

** รายละเอียดการ mount File System เก็บอยู่ใน File /etc/fstab

แสดงเนื้อที่ใน File System ต่างๆ ที่ใช้งานอยู่ด้วยคำสั่ง df

```
# df
Filesystem Size Used Avail Use% Mounted on
/dev/sda1 3.9G 1.7G 2.1G 45% /
```

คำสั่งที่ใช้แสดงสถานะการ mount File System คือคำสั่ง **mount**

```
# mount
/dev/sda1 on / type ext2 (rw)
none on /proc type proc (rw)
none on /proc/bus/usb type usbdevfs (rw)
```

คำสั่ง **mount** ใช้ต่อเชื่อม File System เข้ามาใช้งาน มีรูปแบบของคำสั่งดังนี้

```
mount <Option> <File System> <Mount Point>
```

หากทำการ “mount cdrom” จะมีชื่อของ device เป็น /dev/cdrom ไปไว้ที่ mount point /mnt ทำการตรวจสอบอุปกรณ์ที่ทำการ mount แล้วด้วยคำสั่ง **df**

```
# mount /dev/cdrom /mnt
mount: block device /dev/cdrom is write-protected, mounting
read-only
# ls -l /mnt
# df
Filesystem Size Used Avail Use% Mounted on
/dev/sda1 3.9G 1.8G 1.9G 49% /
/dev/cdrom 695M 695M 0 100% /mnt
```

mount floppy disk (Drive A) มีชื่อ device เป็น /dev/fd0

```
# mount /dev/fd0 /mnt
mount: block device /dev/fd0 is write-protected, mounting readonly
# df
Filesystem Size Used Avail Use% Mounted on
/dev/sda1 3.9G 1.8G 1.9G 49% /
/dev/fd0 1.5M 1.2M 222K 85% /mnt
# ls -l /mnt
```

ยกเลิกการ mount โดยใช้ คำสั่ง **umount** โดย ภายใต้ mount point ต้อง ไม่มี user ใดใช้งานอยู่

```
# umount /mnt
# df
Filesystem Size Used Avail Use% Mounted on
/dev/sda1 3.9G 1.8G 1.9G 49% /
```

ถ้า file system ใดมีรายละเอียดการ mount อยู่ใน file /etc/fstab การ mount ใช้เพียงคำสั่ง mount แล้วตามด้วย mount point เช่น

```
mount /mnt/cdrom
```

```
mount /mnt/floppy
```

File ที่เก็บค่าเริ่มต้นการทำงานของ Linux ที่สำคัญ

File “/etc/inittab”

ใช้กำหนดสถานะของระบบเมื่อระบบ Boot ขึ้นมา แบ่งออกเป็นระดับตั้งแต่ 0-6

0 สถานะการปิดเครื่อง

1 Single User Mode เป็นสถานะที่การเข้าใช้งานเครื่องผ่าน Console ไม่มี

Password สำหรับการ Login

2 Multi User (ไม่มีการใช้งาน Network File System)

3 Multi User มี Network File System (NFS)

4 สถานะนี้ไม่มีการใช้งาน

5 XWindows ทำงานแบบ Graphic Mode

6 สถานะการ Reboot เครื่อง

ข้อมูลสำคัญใน file

```
id:3:initdefault:
```

เป็นการบอกว่าสถานะการทำงานหลังจาก Boot แล้วให้ทำงานที่ runlevel 3 Full Multi User Mode

ใน Runlevel 0 และ 6 ห้ามกำหนดเป็น initdefault เพราะเครื่องจะถูก Shutdown หรือ Reboot ตลอดเวลา ตามปกติแล้ว initdefault จะอยู่ที่ level 3 หรือ level 5

การใช้คำสั่ง **linux single** ขณะที่เครื่องกำลัง boot (ที่ LILO Prompt) เป็นการกำหนดให้ระบบ Boot เข้าสู่ Single User Mode (level 1) เพื่อข้ามขั้นตอนการ Login ใช้ในกรณีที่มี Password สำหรับ Root

File “/etc/fstab”

เป็น file ที่เก็บรายละเอียดการใช้งาน File System ในระบบ มีโครงสร้างดังนี้

<device> *<mount point>* *<type>* *<mount option>* *<dump>* *<fsck>*

/dev/sda1 */ext2* *exec,dev,suid,rw,usrquota* 1 1

none */dev/pts* *devpts* *mode=0620* 0 0

none */mnt/cdrom* *supermount* *dev=/dev/hdc,fs=auto,ro,--,iocharset=iso8859-1,codepage=850,umask=0* 0 0

none */mnt/floppy* *supermount* *dev=/dev/fd0,fs=auto,--,iocharset=iso8859-1,sync,codepage=850,umask=0* 0 0

none */proc* *proc* *defaults* 0 0

/dev/sda5 *swap* *swap* *defaults* 0 0

File /etc/hosts

เก็บชื่อเครื่องและ IP Address ของเครื่อง รูปแบบของข้อมูลใน File
รูปแบบ

<IP Address> *<hostname.domainname>* *<hostname>*

ตัวอย่าง

127.0.0.1 *localhost.localdomain* *localhost*

172.16.80.2 *hplinux.info.com* *hplinux*

File “/etc/resolv.conf”

กำหนดรายละเอียดการติดต่อกับ DNS Server รูปแบบของข้อมูลใน File

search <Domain Name>

nameserver <IP Address DNS Server>

ตัวอย่าง

search info.com

nameserver 172.16.80.2

File “/var/log/messages”

เก็บ System Logging (เหตุการณ์ต่างๆ ที่เกิดขึ้นในระบบ) เวลาเกิดการทำงานผิดพลาดของ Service สามารถเปิดดูรายละเอียดของการทำงานได้จาก file นี้ ตัวอย่างของ System Logging

```
Dec 8 04:12:40 sim named[372]: Lame server on '85.237.144.207.inaddr.
arpa' (in '144.207.IN-ADDR.ARPA?'): [206.74.254.10].53
'DNS2.INFOAVE.NET'
Dec 8 04:12:44 sim named[372]: Lame server on '85.237.144.207.inaddr.
arpa' (in '144.207.IN-ADDR.ARPA?'): [206.74.254.2].53
'DNS4.INFOAVE.NET'
```

คำสั่ง บอกรายละเอียดของ Hardware ของระบบ

dmesg

```
# dmesg
Linux version 2.2.16-22 (root@porky.devel.redhat.com) (gcc version
egcs-2.91.66 19990314/Linux (egcs-1.1.2 release)) #1 Tue Aug 22
16:49:06 EDT 2000
Detected 267277 kHz processor.
Console: colour VGA+ 80x25
Calibrating delay loop... 532.48 BogoMIPS
Memory: 192232k/196608k available (1048k kernel code, 412k reserved,
```

คำสั่งเกี่ยวกับ Network**Startup และ Shutdown Interface**

ifconfig <interfacename> up

ifconfig <interfacename> down

กำหนด IP Address ให้กับ Interface

```
ifconfig <interface name> inet <ip address> netmask <subnetmask> up
```

```
# ifconfig eth0 inet 192.168.1.1 netmask 255.255.255.0 up
# ifconfig eth0
eth0 Link encap:Ethernet HWaddr 00:02:44:0C:C1:1B
inet addr:192.168.1.1 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::202:44ff:fe0c:c11b/10 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:3238 errors:0 dropped:0 overruns:0 frame:0
TX packets:5500 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:295281 (288.3 Kb) TX bytes:3401104 (3.2 Mb)
Interrupt:10 Base address:0xe000
```

คำสั่งสำหรับแสดงสถานะการทำงานของ Network Interface Card และรายละเอียดของ

TCP/IP

```
# ifconfig
eth0 Link encap:Ethernet HWaddr 00:30:6E:0A:06:1B
inet addr:172.16.80.2 Bcast:172.16.255.255 Mask:255.255.0.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:7286620 errors:0 dropped:0 overruns:0 frame:0
TX packets:51041 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:1072907612 (1023.2 Mb) TX bytes:8360895 (7.9 Mb)
Interrupt:26 Base address:0x9000
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:91385 errors:0 dropped:0 overruns:0 frame:0
TX packets:91385 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:6477070 (6.1 Mb) TX bytes:6477070 (6.1 Mb)
```

คำสั่งตรวจสอบสถานะการติดต่อบน TCP/IP

ping <ip address>

```
# ping 172.16.1.254
PING 172.16.1.254 (172.16.1.254) from 172.16.80.2 : 56(84) bytes of data.
64 bytes from 172.16.1.254: icmp_seq=1 ttl=128 time=0.603 ms
64 bytes from 172.16.1.254: icmp_seq=2 ttl=128 time=8.17 ms
64 bytes from 172.16.1.254: icmp_seq=3 ttl=128 time=0.544 ms
64 bytes from 172.16.1.254: icmp_seq=4 ttl=128 time=0.560 ms
(กด Ctrl+c เพื่อจบการทำงาน)
--- 172.16.1.254 ping statistics ---
4 packets transmitted, 4 received, 0% loss, time 3011ms
rtt min/avg/max/mdev = 0.544/2.470/8.176/3.294 ms
```

คำสั่งตรวจสอบการส่งข้อมูลผ่าน Router

tracert <ip address หรือ domain name>

```
# traceroute www.chula.ac.th
traceroute to www1.netserv.chula.ac.th (161.200.192.1), 30 hops max, 38 byte packets
 1 161.200.80.227 (161.200.80.227) 0.883 ms 0.736 ms 0.720 ms
 2 fl-0-2-8510-cen32.it.chula.ac.th (161.200.255.154) 2.374 ms 1.519 ms 1.807 ms
 3 g1-0-0-8540-cen59.it.chula.ac.th (161.200.255.230) 0.996 ms 0.975 ms 0.900 ms
 4 f4-0-7513-cen59.it.chula.ac.th (161.200.255.173) 1.912 ms 1.607 ms 1.865 ms
 5 www1.netserv.chula.ac.th (161.200.192.1) 1.497 ms 1.407 ms 1.370 ms
```

คำสั่งแสดง Routing Table ของ Server ใช้คำสั่ง **route** หรือ **netstat -rn**

```
# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
172.16.0.0 * 255.255.0.0 U 0 0 0 eth0
127.0.0.0 * 255.0.0.0 U 0 0 0 lo
```

การเปิดปิดเครื่อง Server

คำสั่ง **reboot** ใช้สำหรับ Restart เครื่อง Server

```
# reboot
```

คำสั่ง **halt** ใช้สำหรับปิดเครื่อง Server

```
# halt
```

shutdown ใช้สำหรับ shutdown หรือ restart เครื่อง Server (แล้วแต่ Option ที่ใช้)
shutdown -h now เหมือนกับ **halt** ใช้สำหรับสั่งปิดเครื่อง

```
# shutdown -h now
```

shutdown -r now เหมือนกับ **reboot** สั่ง restart เครื่อง

```
# shutdown -r now
```

การ **reboot** เครื่องสามารถทำได้โดยไม่ต้อง login เป็น root ที่ Console กดปุ่ม **Ctrl+Alt+Del**

การจัดการ *User* และ *Group*

การเพิ่ม *Group*

ใช้คำสั่ง `groupadd`

```
groupadd -g <groupid> <groupname>
```

```
#groupadd -g 2000 admin
```

ทำได้โดยเข้าไปเพิ่มข้อมูลในต่อท้าย file `/etc/group`

โดยรูปแบบของ `group` ที่เพิ่มมีรูปแบบดังนี้

```
<Groupname>:::<Groupid>:
```

ตัวอย่างเช่นเพิ่ม `group` ชื่อ `admin` มี `group id` 1000 ทำได้โดยคำสั่ง

```
#pico /etc/group
```

จากนั้นเพิ่มบรรทัดต่อไปนี้ที่ท้าย file

```
admin::1000:
```

การลบ *Group*

```
groupdel <groupname>
```

```
#groupdel admin
```

การเพิ่ม *User*

ใช้คำสั่ง `useradd` รูปแบบของคำสั่งมีดังนี้

```
useradd <Login> -u <User ID> -g <Group> -c FullNamet -d <Home>
```

การกำหนด `password` มีรูปแบบคำสั่งดังนี้

```
passwd <Login>
```

ตัวอย่างให้เพิ่ม “user garfield” มี password เป็น “usr400”

Login = garfield

ค่าอื่นๆที่เหลือ กำหนดตามความเหมาะสมดังนี้

User ID = 1001

Group = users

Full Name = Garfield

Home Directory = /home/garfield

กำหนดให้ Home Directory เป็น /home/garfield directory /home ซึ่งเป็น directory ด้านบนต้อง
ถูกสร้างก่อน ในกรณีที่ directory ด้านบนของ Home Directory ยังไม่ได้ถูกสร้างต้องทำการสร้างด้วย
คำสั่ง

```
mkdir -p <Parent Directory>
```

โดยที่ Parent Directory คือ /home (directory ด้านบนของ Home Directory)

ใช้คำสั่ง useradd เพื่อเพิ่ม user โดยมี option ดังนี้

```
#useradd garfield -u 1001 -g users -c garfield -d /home/garfield
```

กำหนดรหัสผ่านเป็น “usr400” ด้วยคำสั่ง

```
#passwd garfield
```

การลบ User

ใช้คำสั่ง userdel โดยรูปแบบคำสั่งดังนี้

```
userdel -r <username>
```

โดย option -r จะทำการลบ Home Directory และ Mail Box ของ User

ใช้คำสั่ง userdel เพื่อลบ user Garfield

```
#userdel -r !arfield
```

File และ Directory Permission

คำสั่งที่ใช้แสดงค่า permission ของ file หรือ directory

ใช้คำสั่ง `ls -al <ชื่อfileหรือdirectory>`

ตัวอย่าง แสดง permission ของ file `/etc/passwd` ด้วยคำสั่ง `ls -al /etc/passwd`

```
#ls -al /etc/passwd
-rw-r--r-- 1 root root 876 Jan 4 10:20 /etc/passwd
```

ตัวอย่าง แสดง permission ของ directory `/home` ด้วยคำสั่ง `ls -al /home`

```
#ls -al /home
dwxr-xr-x 6 root root 1024 Nov 20 1998 ./
```

ในกรณีของ directory ให้เลือกเฉพาะรายการที่มี column สุดท้าย เป็น `./` ส่วน column แรกของผลลัพธ์ (ของ file `/etc/passwd`) `-rw-r--r--` มี 10 ตัวอักษร

ตัวแรก - เป็นตัวบอกว่าเป็น **file** (ถ้าเป็น directory จะเป็นตัว **d**)

3 ตัวต่อมา `rw-` เป็นตัวบอก permission ของเจ้าของ file

3 ตัวต่อมา `r--` เป็นตัวบอก permission ของ Group

3 ตัวต่อมา `r--` เป็นตัวบอก permission ของคนอื่นๆ

ความหมายของตัวอักษรใน Permission

กรณีเป็น file

r สิทธิในการอ่านเนื้อหาภายใน file

w สิทธิในการเขียน/แก้ไขข้อมูลภายใน file

x สิทธิในการเรียก file ขึ้นมาทำงาน

กรณีเป็น directory

r สิทธิในแสดงรายชื่อ file ภายใน directory

w สิทธิในการเพิ่ม/ลบ/เปลี่ยน ชื่อ file ใน directory

x สิทธิในการเข้าไปใน directory

ทั้งที่เป็น file และ directory

column ที่สอง แสดงเจ้าของ file

column ที่สาม Group ของ file

คำสั่งที่ใช้กำหนด permission

`chmod <mode> <ชื่อfileหรือdirectory>`

โดยค่าของ mode แทนด้วยตัวเลขสามตัว ตัวแรกแทนสิทธิของเจ้าของ ตัวที่สองแทนสิทธิของ Group ตัวที่สามแทนสิทธิของคนอื่นๆดังนี้

r = 4, w = 2, x = 1, - = 0

ถ้าต้องการเปลี่ยนสิทธิของ file “/tmp/test.txt” เป็น **rxw-rw-rw-** ส่วนของ mode จะเป็นดังนี้

$mode = r+w+x \ r+w \ r+w = 4+2+1 \ 4+2 \ 4+2 = 7 \ 6 \ 6$

กำหนด permission ด้วยคำสั่ง `chmod` ตามนี้

```
$ touch /tmp/test.txt
```

```
$ ls -l /tmp/test.txt
```

```
-rw-r--r-- 1 mong users 0 Jan 14 09:39 /tmp/test.txt
```

```
$ chmod 766 /tmp/test.txt
```

```
$ ls -l /tmp/test.txt
```

```
-rwxrw-rw- 1 mong users 0 Jan 14 09:39 /tmp/test.txt
```

การจัดการแพ็คเกจบน Ubuntu

การจัดการแพ็คเกจหรือเรียกให้เข้าใจง่ายๆ คือระบบติดตั้งโปรแกรม หากเป็นระบบปฏิบัติการวินโดวส์เราสามารถติดตั้งโปรแกรมด้วยการดับเบิลคลิกไฟล์นามสกุล .exe หรือ .msi เช่น setup.exe ที่เรารู้จักคุ้นเคยกัน

ลินุกซ์กำเนิดมาจากภาษาซี โปรแกรมที่ติดตั้งบนลินุกซ์ส่วนใหญ่ก็เขียนมาจากภาษาซีด้วย ถ้าไม่มีระบบการติดตั้ง เราก็ต้องติดตั้งโปรแกรมต่างๆ จากซอร์สโค้ดด้วยการคอมไพล์ด้วยสามคำสั่งหลักได้แก่ ./configure; make; make install ตามลำดับ ซึ่งยุ่งยากต่อผู้ใช้งานทั่วไป ผู้พัฒนาจึงต้องนำโปรแกรมต่างๆ เหล่านี้ให้ติดตั้งได้ง่ายโดยการนำโปรแกรมต่างๆ เหล่านี้แปลงให้เป็น .deb ซึ่งเป็นไบนารีไฟล์ (คล้าย .exe) สามารถติดตั้งได้เลยโดยไม่ต้องคอมไพล์อีก ซึ่งจะเรียกว่า แพ็คเกจ (package)

Debian ได้เป็นผู้นำที่ทำให้เกิด APT (Advanced Packaging Tool) ซึ่งเป็นระบบการติดตั้งแพ็คเกจที่มีประสิทธิภาพตัวหนึ่ง ซึ่ง Ubuntu พัฒนามาจาก Debian ระบบการติดตั้งแพ็คเกจก็ยังคงใช้ APT ซึ่งผู้ใช้ Debian คุ้นเคยและชอบเรียกกันว่า แอปเก็ต หรือ apt-get การติดตั้งโดยใช้ apt-get นั้นเป็นการติดตั้งแบบออนไลน์ นั่นก็คือ ดาวน์โหลด และติดตั้งโปรแกรมผ่าน Server ที่ให้บริการ การจัดการแพ็คเกจบน Ubuntu ไม่ได้มีแต่การติดตั้งแบบออนไลน์เท่านั้น การติดตั้งแบบออฟไลน์ก็ทำได้เช่นกัน

การติดตั้งแพ็คเกจแบบออฟไลน์ด้วย dpkg

การติดตั้งแพ็คเกจแบบออฟไลน์คือการติดตั้งไฟล์ .deb ที่อยู่ในซีดีหรือไฟล์ที่เราดาวน์โหลดมา ซึ่งเราจะติดตั้งไฟล์เหล่านั้นด้วยคำสั่ง dpkg ซึ่งคล้ายกับการใช้คำสั่ง rpm บนลินุกซ์ตระกูล Red Hat เมื่อเราติดตั้งโปรแกรมด้วย apt-get install เช่น apt-get install rcconf หรือ apt-get install postfix ไฟล์ที่เราดาวน์โหลดมาก็จะถูกเก็บไว้ที่ /var/cache/apt/archive ซึ่งสามารถนำมาติดตั้งแบบออฟไลน์ด้วยคำสั่ง dpkg ได้ ก่อนที่จะติดตั้งลองมาดูหน้าตา และทำความเข้าใจเกี่ยวกับชื่อไฟล์เหล่านี้ก่อน

```
postfix_2.5.1-2ubuntu1.1_amd64.deb
```

```
rcconf_1.19_all.deb
```

ที่เห็นทั้งหมดจะเป็นชื่อไฟล์ที่นามสกุล .deb ในชื่อไฟล์นี้มีอะไรแฝงอยู่บ้าง

```
<PackageName>_<VersionNumber>-<UbuntuRevisionNumber>_<UbuntuArchitecture>.deb
```

ตัวอย่าง

ไฟล์ postfix_2.5.1-2ubuntu1.1_amd64.deb

ชื่อแพ็คเกจ postfix

เวอร์ชัน 2.5.1-2

UbuntuRevisionNumber 1.1
ใช้กับสถาปัตยกรรมซีพียู amd64

ไฟล์ rcconf_1.19_all.deb

ชื่อแพ็คเกจ rcconf

เวอร์ชัน 1.19

ใช้กับสถาปัตยกรรมซีพียู ทุกประเภท

<code>dpkg -c <.deb file></code>	แสดงไฟล์ทั้งหมดที่จะติดตั้งโดยไฟล์ .deb
<code>dpkg -I <.deb file></code>	แสดงข้อมูลของไฟล์ .deb
<code>dpkg -p <package></code>	แสดงข้อมูลเกี่ยวกับแพ็คเกจ
<code>dpkg -S <filename></code>	ต้องการรู้ว่าไฟล์เป็นส่วนประกอบของแพ็คเกจใด
<code>dpkg -l</code>	แสดงรายชื่อแพ็คเกจทั้งหมดที่ได้ติดตั้งไปแล้ว
<code>dpkg -L <package></code>	แสดงไฟล์ทั้งหมดที่ถูกติดตั้งไปจากแพ็คเกจนั้นๆ
<code>dpkg -s <package></code>	แสดงสถานะของแพ็คเกจ
<code>sudo dpkg -i <.deb file></code>	ติดตั้งไฟล์ .deb
<code>sudo dpkg -r <package></code>	ลบแพ็คเกจ แต่ยังเก็บไฟล์ที่ดาวน์โหลด
<code>sudo dpkg -P <package></code>	ลบแพ็คเกจ และคอนฟิกไฟล์ที่ถูกสร้างโดยแพ็คเกจนั้นๆ
<code>sudo dpkg -x <.deb file> <directory></code>	ขยายไฟล์ .deb ไปยังไดเรกทอรีที่ต้องการ

ตาราง การใช้งานคำสั่ง dpkg

การติดตั้งแพ็คเกจแบบออนไลน์ ด้วย *apt-get*

การติดตั้งแบบออนไลน์คือการดาวน์โหลด และติดตั้งแพ็คเกจจากเซิร์ฟเวอร์ที่ให้บริการ หรือจาก CD แต่ไม่ได้เข้าถึงไฟล์ .deb โดยตรง เหมือน dpkg ส่วนประกอบสำคัญที่ทำให้เราสามารถติดตั้งโปรแกรมต่างๆ ได้นั้นก็คือไฟล์ที่ระบุ Server ปลายทางที่ให้บริการ นั่นก็คือไฟล์ /etc/apt/sources.list ซึ่งจะมีมาให้แล้วตอนที่เราติดตั้ง ubuntu ก่อนที่จะใช้งานหรือติดตั้งแพ็คเกจใดควรจะเรียกใช้คำสั่ง `apt-get update` เพื่ออัปเดตฐานข้อมูลของแพ็คเกจให้เป็นปัจจุบัน

ในการใช้งาน apt-get install เป็นการดาวน์โหลดไฟล์จากเซิร์ฟเวอร์ที่ให้บริการ หรือจาก CD ติดตั้ง ทั้งนี้ขึ้นอยู่กับไฟล์ /etc/apt/sources.list สำหรับการดาวน์โหลดไฟล์ .deb จะมีที่เก็บไฟล์อยู่ที่ /var/cache/apt/archives/ ซึ่งสามารถเก็บเอาไว้ติดตั้งด้วยคำสั่ง dpkg ได้ หากแพ็คเกจที่เราได้ติดตั้ง และลบออกไปแล้ว และต้องการที่จะติดตั้งใหม่อีกครั้งถ้ายังมีไฟล์อยู่ใน /var/cache/apt/archives/ ก็จะสามารถติดตั้งแพ็คเกจด้วยคำสั่ง dpkg ได้ทันที โดยไม่ต้องใช้คำสั่ง apt-get

sudo apt-get update	อัปเดตฐานข้อมูลของแพ็คเกจทั้งหมด รันคำสั่งนี้เมื่อ เปลี่ยนแปลงไฟล์ /etc/apt/sources.list
sudo apt-cache search <keyword>	ค้นหาแพ็คเกจที่ต้องการติดตั้ง
sudo apt-get install <package>	ดาวน์โหลด และติดตั้งแพ็คเกจ
sudo apt-get remove <package>	ลบแพ็คเกจที่ติดตั้งไปแล้ว
sudo apt-get -d install <package>	ดาวน์โหลดแพ็คเกจเก็บไว้ใน /var/cache/apt/archives แต่ไม่ติดตั้ง
sudo apt-cache show <package>	แสดงข้อมูลรายละเอียดของแพ็คเกจ
sudo apt-get upgrade	เช็คว่าแพ็คเกจที่ติดตั้งไปทั้งหมดมีแพ็คเกจใดมีการอัปเดตบ้าง ต่อจากนั้นก็จะถามว่าต้องการดาวน์โหลดและติดตั้งหรือไม่
sudo apt-get dist-upgrade	อัปเดตระบบทั้งหมด เช่นการเปลี่ยนเวอร์ชันจาก 7.10 เป็น 8.04
sudo apt-get autoclean	ลบแพ็คเกจที่ดาวน์โหลดมาไม่สมบูรณ์ หรือแพ็คเกจที่ไม่ได้ใช้ติดตั้งแล้ว
sudo apt-get clean	ลบทุกแพ็คเกจที่ดาวน์โหลดมาเก็บอยู่ที่ /var/cache/apt/archives
sudo apt-get -f install	กรณีที่มีความผิดพลาดเกิดขึ้นในขณะที่กำลังติดตั้งแพ็คเกจ เช่น สั่งยกเลิก หรือ ไฟฟ้าดับ แล้วใช้คำสั่ง apt-get install แล้วมีข้อความ error สามารถใช้คำสั่ง apt-get -f install ช่วยแก้ปัญหาได้
sudo apt-config -v	แสดงเวอร์ชัน และรายละเอียดของ APT utilities
sudo apt-cache stats	แสดงสถิติของแพ็คเกจที่ติดตั้งไปแล้วทั้งหมด

sudo apt-cache depends	ดูแพ็คเกจที่ขึ้นต่อกันของแพ็คเกจที่จะติดตั้ง
sudo apt-cache pkgnames	แสดงแพ็คเกจทั้งหมดที่ได้ติดตั้งไป

ตาราง การใช้งานคำสั่ง apt-get

การติดตั้งแพ็คเกจแบบออนไลน์ด้วย *aptitude*

เป็นการติดตั้งแพ็คเกจแบบออนไลน์เหมือน *apt-get* แต่ *aptitude* สามารถใช้งานได้ ในแบบโปรแกรม และคำสั่ง ถ้าใช้งานแบบโปรแกรมก็เป็นลักษณะดังรูปที่ 1



รูป การติดตั้งแพ็คเกจด้วย aptitude

ความสามารถพิเศษของ aptitude ที่มากกว่า apt-get คือ การติดตามแพ็คเกจกำพว้า (orphaned package) หรือแพ็คเกจที่ไม่ได้ใช้แล้ว แล้วจะลบให้อัตโนมติ ในการติดตั้งแพ็คเกจจะมีแพ็คเกจที่ขึ้นต่อกัน(dependency) กับแพ็คเกจอื่น เช่น ติดตั้งแพ็คเกจ a แต่การที่จะติดตั้งแพ็คเกจ a ได้ นั้น ต้องติดตั้ง b และ c ด้วย เราจะเรียก a ว่าแพ็คเกจหลัก เรียก b และ c ว่าแพ็คเกจที่ขึ้นต่อกัน โดยปกติถ้าลบแพ็คเกจ a ออกไป b และ c อาจจะไม่ถูกลบออกถ้าเราใช้คำสั่ง apt-get remove a ซึ่ง b และ c ก็ยังคงค้างอยู่ในเครื่อง ถ้าใช้ aptitude remove a จะทำให้ b และ c ก็จะถูกลบไปด้วย ถ้า b และ c ไม่มีแพ็คเกจใดใช้

sudo aptitude	เข้าใช้งาน โปรแกรม Ctrl+t เข้าสู่เมนู กดปุ่ม q เพื่อออกจากโปรแกรม
sudo aptitude help	คู่มือการใช้งาน aptitude
sudo aptitude search <keyword>	ค้นหาแพ็คเกจ
sudo aptitude update	อัปเดตแพ็คเกจที่มีอยู่
sudo aptitude upgrade	อัปเดตทุกแพ็คเกจที่ติดตั้งไปแล้วเป็นเวอร์ชันล่าสุด
sudo aptitude show <package>	แสดงข้อมูลของแพ็คเกจว่าได้ติดตั้งไปแล้วหรือไม่
sudo aptitude download <package>	ดาวน์โหลดแพ็คเกจมาอย่างเดียวแต่ไม่ติดตั้ง
sudo aptitude clean	ลบไฟล์ .deb ที่ดาวน์โหลดมาที่เก็บอยู่ที่ /var/cache/apt/archives
sudo aptitude autoclean	ลบไฟล์ .deb ที่ไม่ได้ใช้งานใน /var/cache/apt/archives
sudo aptitude install <package>	ติดตั้งแพ็คเกจ sudo aptitude remove <package> ลบแพ็คเกจที่ติดตั้งไปแล้ว
sudo aptitude dist-upgrade	อัปเดตระบบเป็น Ubuntu เวอร์ชันล่าสุด เช่น ใช้ ubuntu 7.10 แล้วต้องการอัปเดตเป็น >ubuntu 8.10

ตาราง การใช้งานคำสั่ง aptitude

apt-get หรือ aptitude ผ่าน proxy authentication

ปัจจุบันเมื่อหลายหน่วยงานทำตามพระราชบัญญัติว่าด้วยการกระทำความผิดทาง คอมพิวเตอร์ พ.ศ. 2550 หลายหน่วยงานก็ต้องใช้ proxy authentication ซึ่งต้องระบุ username และรหัสผ่านก่อนใช้งาน เช่นเดียวกันสำหรับการติดตั้งแบบออนไลน์ ด้วย apt-get หรือ aptitude ก็จำเป็นต้องใช้งานผ่าน proxy authentication เช่นกัน ซึ่งเราต้องไปแก้ไขไฟล์ /etc/apt/apt.conf โดยเพิ่มบรรทัด

```
Acquire::http::Proxy "http://apt:apt123@192.168.2.254:8080";
```

รูปแบบคือ

```
Acquire::http::Proxy "http://username:password@proxy-host:proxy-port";
```

การ Config Server ต่างๆ บน Ubuntu

การกำหนด *IP Address*

ไฟล์ที่ใช้งานเก็บอยู่ที่ `/etc/network/interfaces`

ใช้ text editor เพื่อแก้ไขในที่นี่ใช้ nano เพื่อแก้ไข

รูปแบบคำสั่ง

`nano /etc/network/interfaces`

```

auto lo

iface lo inet loopback

# The primary network interface

auto eth0

iface eth0 inet static

    address xxx.xxx.xxx.xxx

    netmask xxx.xxx.xxx.xxx

    gateway xxx.xxx.xxx.xxx

```

address คือ IP Address ของเครื่อง Server

netmask คือ Subnet Mask ของ Network

gateway คือ IP Address ของ Router

หากต้องการ *Setup interface* เป็น *dhcp* ทำการแก้ไขดังนี้

```

auto eth0

iface eth0 inet dhcp

```

การกำหนด *DNS Server*

ไฟล์ที่ใช้งานเก็บอยู่ที่ `/etc/resolv.conf`

ใช้ text editor เพื่อแก้ไขในที่นี่ใช้ nano เพื่อแก้ไข

รูปแบบคำสั่ง

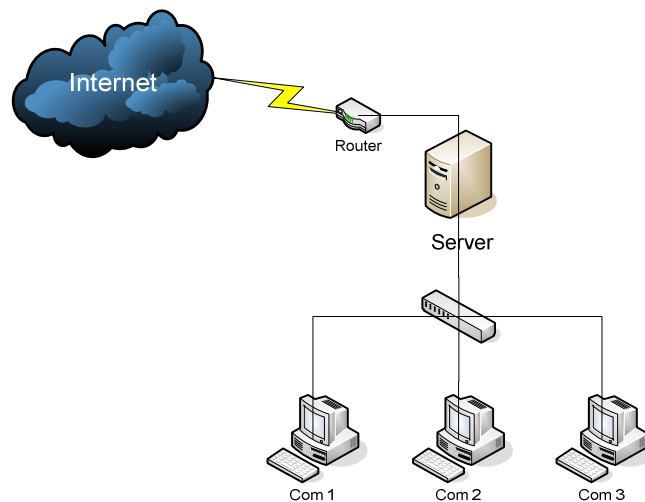
`nano /etc/resolv.conf`

```
search lamptech.ac.th
```

```
nameserver xxx.xxx.xxx.xxx * IP Address ของ nameserver ตัวที่ 1
```

```
nameserver xxx.xxx.xxx.xxx * IP Address ของ nameserver ตัวที่ 2
```

การทำ IP Forward



1. ตรวจสอบก่อนว่า IP Forwarding ได้ถูก enabled ไว้แล้วหรือยัง

ใช้คำสั่ง `sysctl net.ipv4.ip_forward` ถ้าแสดงผลขึ้นมา `net.ipv4.ip_forward=0` หมายถึงยังไม่ได้ enabled หรือจะใช้คำสั่ง `cat /proc/sys/net/ipv4/ip_forward` ก็ได้

2. Enable IP Forwarding

ใช้คำสั่ง `sudo sysctl -w net.ipv4.ip_forward=1` หรือ `echo 1 > /proc/sys/net/ipv4/ip_forward` จากนั้น restart process ด้วยคำสั่ง `/etc/init.d/procps.sh restart` แต่ถ้าต้องการเปลี่ยนค่าถาวร ไปเพิ่ม `net.ipv4.ip_forward=1` ลงใน `/etc/sysctl.conf` เมื่อเปิดเครื่องคราวหน้าจะได้ไม่ต้องทำซ้ำๆ อีก

3. ทำ NAT ซึ่งเป็นขั้นตอนสุดท้าย ใช้คำสั่ง iptables ในการทำ address translation ผ่าน interface eth0

ใช้คำสั่ง `sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE`

4. สั่ง Restart Service

`#/etc/init.d/networking restart`

การติดตั้งและคอนฟิกเนมเซิร์ฟเวอร์ ในองค์กร

ในที่นี้จะกล่าวถึงการติดตั้งและใช้งาน Name server ในองค์กรเท่านั้นจะไม่กล่าวถึงระบบโดเมนของอินเทอร์เน็ต ถามว่าเนมเซิร์ฟเวอร์ในองค์กรไม่มีได้มัย ตอบว่าได้ แต่จะใช้งานได้เฉพาะ IP เท่านั้น เช่น ติดตั้งเว็บเซิร์ฟเวอร์ในองค์กร มี IP 192.168.2.1 เวลาใช้งานจากเบรเซอร์ก็ต้องเรียก `http://192.168.2.1` หากเป็นผู้ดูแลระบบคงไม่มีปัญหาที่จะใช้งานแบบนี้ แต่ผู้ใช้งานทั่วไปคงลำบาก หรือผู้ดูแลระบบได้ติดตั้ง mail sever โดยที่ mail server มี IP address 192.168.2.2 ถ้าไม่มีเนมเซิร์ฟเวอร์ หากส่ง mail ไปยัง User ของสมชาย ก็จะต้องใช้ E-mail Address ดังนี้ `somchai@192.168.2.2` ได้ จากที่กล่าวมาจะเห็นว่า nameserver มีความสำคัญไม่น้อย ถ้าเราเริ่มต้นด้วย Name server แล้ว อย่างอื่นก็จะสะดวกตามไปด้วย ความต้องการในบทนี้

IP Address	ชื่อ
192.168.2.1	www.myintranet.org
192.168.2.2	mail.myintranet.org
192.168.2.10	ns.myintranet.org

โปรแกรม BIND9

โปรแกรมที่เราติดตั้งเพื่อทำเนมเซิร์ฟเวอร์คือ Bind เวอร์ชัน 9 ซึ่งจะเรียกว่า BIND9 BIND (Berkeley Internet Name Domain) โปรแกรม BIND ได้พัฒนาขึ้นที่ University of California at Berkeley เป็นโปรเจกของ นักศึกษาปริญญาตรี ภายใต้การสนับสนุนโดย US Defense Advanced Research Projects Administration (DARPA) โปรแกรม BIND ตั้งแต่พัฒนาจนถึงเวอร์ชัน 4.8.3 ได้รับการดูแลจาก Computer Systems Research Group (CSRG) ที่ UC Berkeley ทีมพัฒนาได้แก่ Douglas Terry, Mark Painter, David Riggle และ Songnian Zhou หลังจากนั้นก็ได้มีการรับช่วงการพัฒนาหลายคนหลายองค์กร จนปัจจุบัน BIND เวอร์ชัน 9 ซึ่งออกมา เมื่อเดือนกันยายน ปี 2000

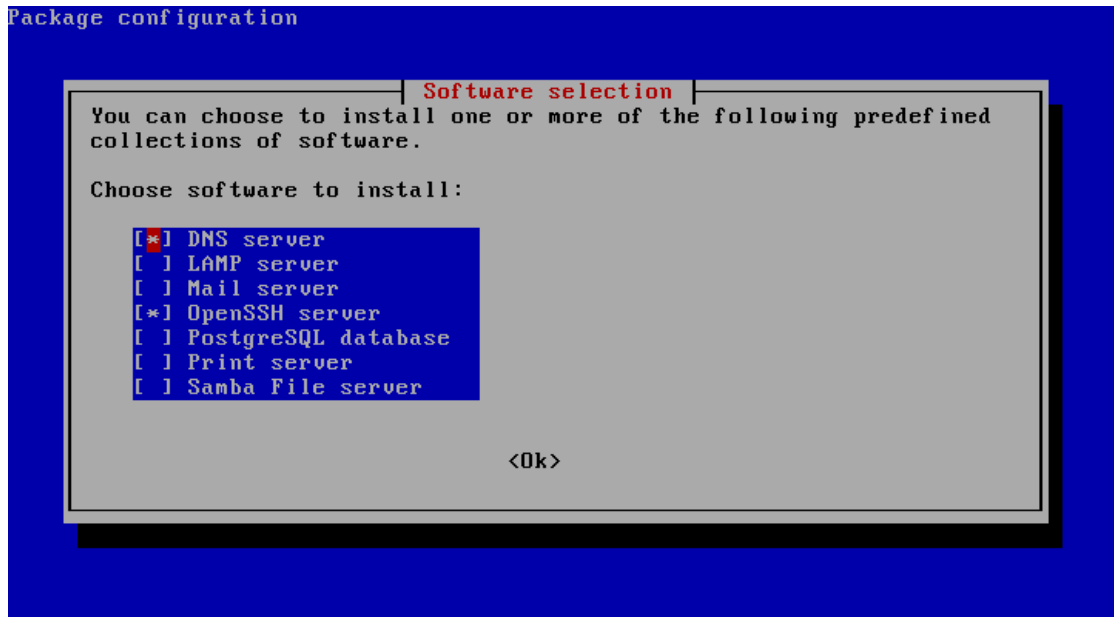
มีหลายวิธีที่จะคอนฟิก BIND9 ส่วนใหญ่เราจะทำเป็น name server แบบ Primary master และ secondary master แบบ Primary Master จะอ่านข้อมูลจากไฟล์คอนฟิกบนเครื่องตัวเอง แบบ Secondary Master จะอ่านข้อมูลจากเครื่องอื่นที่ได้รับอนุญาต

การติดตั้ง BIND9 บน Ubuntu Server

Bind9 อยู่ในแผ่นติดตั้ง Ubuntu Server อยู่แล้ว เราสามารถติดตั้งได้ด้วยคำสั่ง `tasksel`

```
root@ubuntuserver:~#tasksel
```

หรือ `sudo tasksel` ก็แล้วแต่ว่าใครล็อกอินเข้ามาด้วย user ไหน เมื่อเรียกคำสั่ง `tasksel` ก็จะได้ดังภาพ



หรือจะติดตั้งด้วย apt-get

```

root@ubuntuserver:~# aptget
install bind9
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
bind9doc
resolvconf
The following NEW packages will be installed: bind9
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0B/267kB of archives.
After this operation, 762kB of additional disk space will be used.
Selecting previously deselected package bind9.
(Reading database ... 15320 files and directories currently installed.)
Unpacking bind9 (from .../bind9/bind9_9.4.210_i386.deb) ...
Setting up bind9 (1:9.4.210)
...
Adding group `bind' (GID 113) ...

```

```

Done.
Adding system user `bind' (UID 105) ...
Adding new user `bind' (UID 105) with group `bind' ...
Not creating home directory `/var/cache/bind'.
wrote key file "/etc/bind/rndc.key"
Reloading AppArmor profiles : done.
Starting domain name service... bind [ OK ]

```

หลังจากที่ติดตั้ง BIND9 ไปแล้ว ไดรกทอรีที่เก็บไฟล์คอนฟิก ได้แก่

/etc/bind ไฟล์คอนฟิกหลักคือ **/etc/bind/named.conf**

include ไฟล์ หรือ option เพิ่มเติม จะเก็บไว้ที่ **/etc/bind/named.conf.options**

/etc/bind/db.root เก็บ name server ทั่วโลก

Caching Nameserver

Caching Nameserver เป็น Server ตัวถัดไปที่จะให้ลูกข่ายค้นหาหาใน server ตัวแรกไม่เจอ ส่วนใหญ่เราจะใส่ค่า Nameserver ของ ISP ที่เราใช้งานอยู่ เช่น KSC 203.113.24.199, TOT 203.113.24.199 ซึ่ง option นี้อยู่ใน **/etc/bind/named.conf.options**

```

// forwarders {
// 0.0.0.0;
// };

```

เปลี่ยนเป็น

```

forwarders {
203.155.33.1;
203.113.24.199;
};

```

Primary Master Name Server

ในที่นี้จะกล่าวถึงการติดตั้งเนมเซิร์ฟเวอร์ที่เป็น Primary Master คือการอ่านค่าคอนฟิกจากเครื่องตัวเองซึ่งค่าคอนฟิกนี้คือ การเปลี่ยนชื่อเป็น IP Address และเปลี่ยน IP Address เป็นชื่อ สามารถทำได้โดยการ ก๊อปปี้ไฟล์คอนฟิกที่มีอยู่แล้วเอามาแก้ไข ไม่ต้องพิมพ์เองทั้งหมด

1. ก๊อปปี้ไฟล์ด้วยคำสั่ง `root@ubuntu-server:~# cp /etc/bind/db.local /etc/bind/db.myintranet.org` ไฟล์นี้ทำหน้าที่เพื่อเปลี่ยนชื่อเป็น IP Address
2. แก้ไขไฟล์ `/etc/bind/named.conf.local` ด้วย Text Editor ที่ถนัด โดยเพิ่มรายละเอียด ด้านล่างนี้เข้าไป

```
zone "myintranet.org" {
    type master;
    file "/etc/bind/db.myintranet.org";
};
```

การแก้ไขไฟล์ `/etc/bind/db.myintranet.org` ดังนี้

```

;
; BIND data file for local loopback interface
;
$TTL 604800
@      IN      SOA  ns.myintranet.org.  root.myintranet.org. (
1              ; Serial
604800         ; Refresh
86400          ; Retry
2419200       ; Expire
604800 )      ; Negative Cache TTL
@      IN      MX   10   mail.myintranet.org.
;
@      IN      NS   ns.myintranet.org.
@      IN      A    192.168.2.10
ns     IN      A    192.168.2.10
www    IN      A    192.168.2.1
mail   IN      A    192.168.2.2
```

4. ก๊อปปี้ไฟล์ด้วยคำสั่ง `root@ubuntu-server:~# sudo cp /etc/bind/db.127 /etc/bind/db.192` ไฟล์นี้ทำหน้าที่ เปลี่ยนชื่อ IP Address เป็นชื่อ
5. หลังจากนั้นแก้ไขไฟล์ `/etc/bind/named.conf.local`

เพิ่ม

```
zone "2.168.192.inaddr. arpa" {
    type master;
    notify no;
    file "/etc/bind/db.192";
};
```

6. แก้ไขไฟล์ /etc/bind/db.192

```
;
; BIND reverse data file for local loopback interface
;
$TTL 604800
@      IN      SOA    ns.myintranet.org.    root.myintranet.org. (
1              ; Serial
604800        ; Refresh
86400         ; Retry
2419200      ; Expire
604800 )      ; Negative Cache TTL
;
@      IN      NS     ns.
10     IN      PTR    ns.myintranet.org.
1      IN      PTR    www.myintanet.org.
2      IN      PTR    mail.myintanet.org.
```

7. สั่ง restart service ด้วยคำสั่ง `root@ubuntu-server:~# /etc/init.d/bind9 restart`

```
root@ubuntuserver:~#/etc/init.d/networking restart
* Reconfiguring network interfaces... [ OK ]
root@ubuntuserver:~#
```

8. แก้ไขไฟล์ `/etc/resolv.conf`

```
nameserver localhost
```

หรือ

```
nameserver 192.168.2.10
```

9. แล้ววีสดำเนินการด้วยคำสั่ง `root@ubuntu:~# /etc/init.d/networking restart`
ทดสอบการทำงานของเซิร์ฟเวอร์ บนเซิร์ฟเวอร์ด้วยคำสั่ง `dig`

```
root@ubuntu:~/etc/bind# dig myintranet.org

;<<>> DiG 9.4.2 <<>> myintranet.org
;; global options: printcmd
;; Got answer:
;; >>HEADER<<opcode: QUERY, status: NOERROR, id: 38927
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;myintranet.org.      IN      A

;; ANSWER SECTION:
myintranet.org.     604800 IN      A       192.168.2.10
;; AUTHORITY SECTION:
myintranet.org.     604800 IN      NS      ns.myintranet.org.
;; ADDITIONAL SECTION:
ns.myintranet.org. 604800 IN      A       192.168.2.10

;; Query time: 1 msec
;; SERVER: 192.168.2.10#53(192.168.2.10)
;; WHEN: Fri Sep 5 17:17:04 2008
;; MSG SIZE rcvd: 81
```

```
root@ubuntuserver:/etc/bind# dig www.myintranet.org

;<<>> DiG 9.4.2 <<>> www.myintranet.org
;; global options: printcmd
;; Got answer:
;>>HEADER<<opcode:QUERY, status:NOERROR, id:1814
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; QUESTION SECTION:
www.myintranet.org.      IN      A
;; ANSWER SECTION:
www.myintranet.org.    604800 IN      A      192.168.2.1
;; AUTHORITY SECTION:
myintranet.org.        604800 IN      NS     ns.myintranet.org.
;; ADDITIONAL SECTION:
ns.myintranet.org.     604800 IN      A      192.168.2.10
;; Query time: 1 msec
;; SERVER: 192.168.2.10#53(192.168.2.10)
;; WHEN: Fri Sep 5 17:17:34 2008
;; MSG SIZE rcvd: 85
```

```

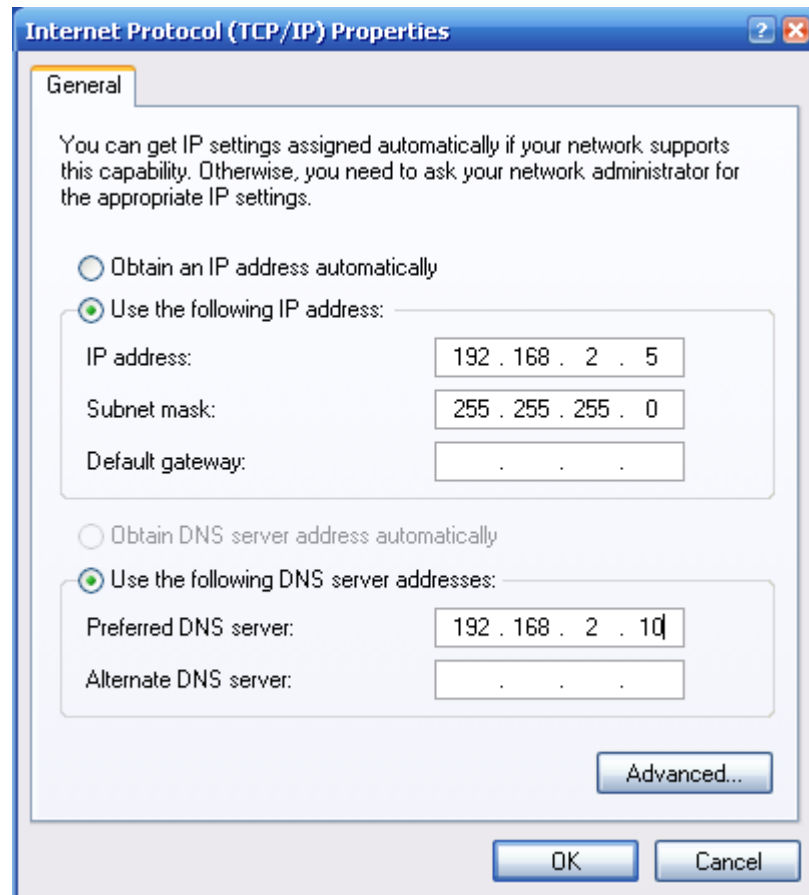
root@ubuntuserver:/etc/bind# dig mail.myintranet.org

;<<>> DiG 9.4.2 <<>> mail.myintranet.org
;; global options: printcmd
;; Got answer:
;>>HEADER<<opcode:QUERY, status:NOERROR, id: 3851
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; QUESTION SECTION:
mail.myintranet.org.      IN      A
;; ANSWER SECTION:
mail.myintranet.org.     604800 IN      A      192.168.2.2
;; AUTHORITY SECTION:
myintranet.org.         604800 IN      NS     ns.myintranet.org.
;; ADDITIONAL SECTION:
ns.myintranet.org.      604800 IN      A      192.168.2.10
;; Query time: 1 msec
;; SERVER: 192.168.2.10#53(192.168.2.10)
;; WHEN: Fri Sep 5 17:19:06 2008
;; MSG SIZE rcvd: 86

```

ถ้าใช้คำสั่ง dig แล้วได้คำตอบข้างต้นแสดงว่าเนมเซิร์ฟเวอร์ทำงานถูกต้องแล้ว

11. ทดสอบการทำงานของเนมเซิร์ฟเวอร์จากเครื่องลูกข่ายที่เป็นลินุกซ์ทำเช่นกัน กับข้อ 8-10
12. ทดสอบการทำงานของเนมเซิร์ฟเวอร์จากเครื่องลูกข่ายวินโดวส์เราชี้ DNS server ไปยัง IP Name Server
ของเรา



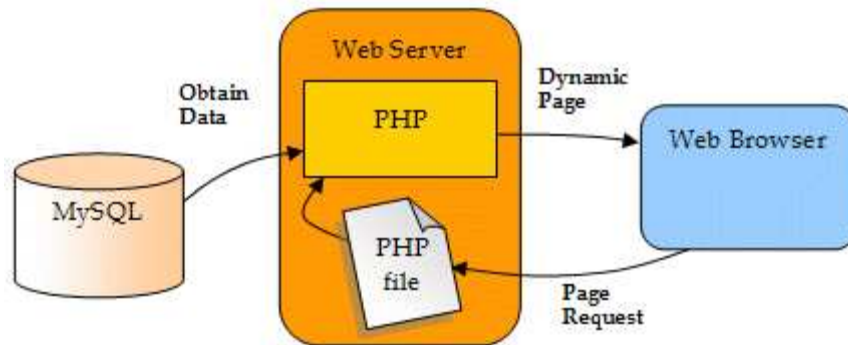
เว็บเซิร์ฟเวอร์ในองค์กร

ทุกวันนี้ทุกคนคงรู้จักเว็บเพจ และท่องอินเทอร์เน็ตค้นหาข้อมูลโดยการเล่นเว็บ จะมีสักกี่คนที่รู้ว่าเบื้องหลังของผู้ให้บริการนั้นมีอะไรอยู่เบื้องหลัง ที่เราเรียกกันว่าเว็บเซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์บนโลกอินเทอร์เน็ต หรือในองค์กรก็คงไม่แตกต่างกันในด้านโปรแกรมที่นำมาใช้งาน เว็บเซิร์ฟเวอร์ของ Linux ส่วนมากใช้ LAMP ย่อมาจาก **L**inux, **A**pache, **M**ySQL และ **P**HP จะเห็นว่าโปรแกรมที่นำมาใช้งานร่วมกันมีถึง 3 โปรแกรมไม่นับ Linux เนื่องจากเว็บในปัจจุบันไม่ได้มีแต่ html ไฟล์เท่านั้น ที่เราเห็นเว็บในปัจจุบันส่วนใหญ่ ทำงานโดยการดึงข้อมูลในฐานข้อมูลมาแสดง โดยผ่านภาษา Script ไม่ว่าจะเป็น PHP, ASP หรืออื่นๆ สำหรับบนลินุกซ์โปรแกรมสุดฮิตเห็นจะเป็น

Apache เป็นเว็บเซิร์ฟเวอร์

MySQL โปรแกรมฐานข้อมูล

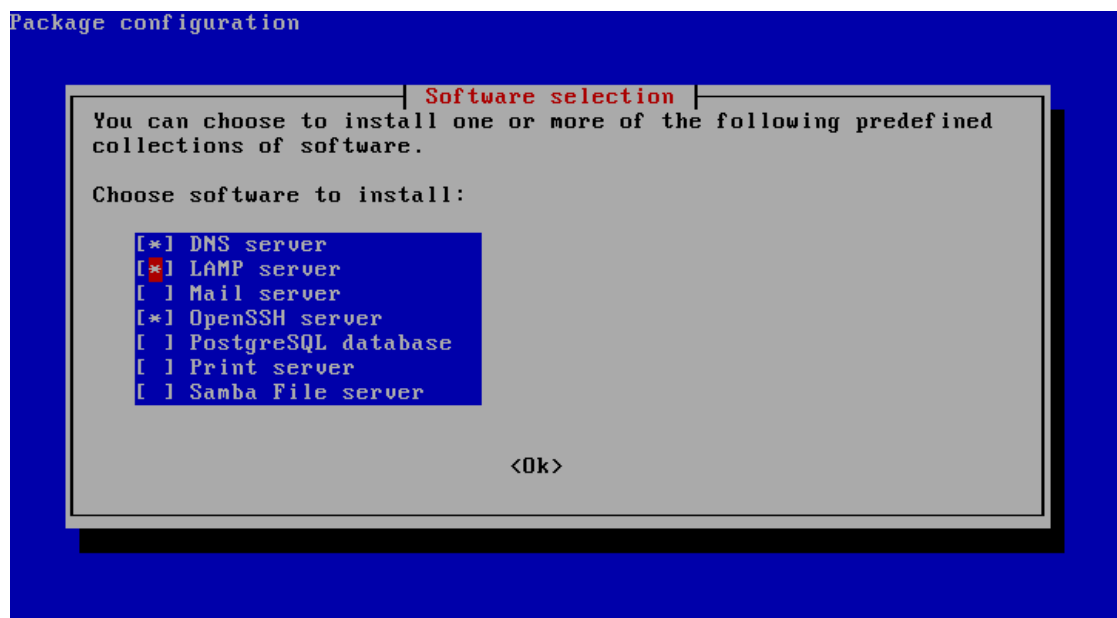
PHP ภาษา Script ทำหน้าที่ติดต่อกับฐานข้อมูล และทำงานร่วมกับ Apache เพื่อนำข้อมูลในฐานข้อมูลมาแสดงผลบนหน้าเว็บ



เครดิตภาพ <http://www.databasejournal.com/img/phpmysql-ch4-fig1.gif>

การติดตั้ง LAMP server

ถ้าหากยังไม่ได้ติดตั้ง LAMP server ตั้งแต่ขั้นตอนติดตั้ง Ubuntu server เราสามารถติดตั้งโดยใช้คำสั่ง `root@webserver:/var/www# tasksel` ก็จะได้ดังภาพ กด Space bar ที่หน้า LAMP server แล้วกด Tab ไปยัง Ok เพื่อเริ่มทำการติดตั้ง



หลังจากนั้นก็ให้กรอกรหัสผ่านของผู้ใช้ที่จะใช้งาน MySQL คือผู้ใช้ root แต่เป็นคนละผู้ใช้กับ root ที่เป็นผู้ใช้ที่มีสิทธิ์สูงสุดในระบบ

Package configuration

Configuring mysql-server-5.0

While not mandatory, it is highly recommended that you set a password for the MySQL administrative "root" user.

If that field is left blank, the password will not be changed.

New password for the MySQL "root" user:

<Ok>

ยืนยันรหัสผ่านอีกครั้ง

Package configuration

Configuring mysql-server-5.0

Repeat password for the MySQL "root" user:

<Ok>

ระบบจะทำการติดตั้งแพ็คเกจอื่นๆจนสำเร็จ

```
Package configuration

Installed php5-mysql | Installing packages |
-----|-----|
| 100% |
```

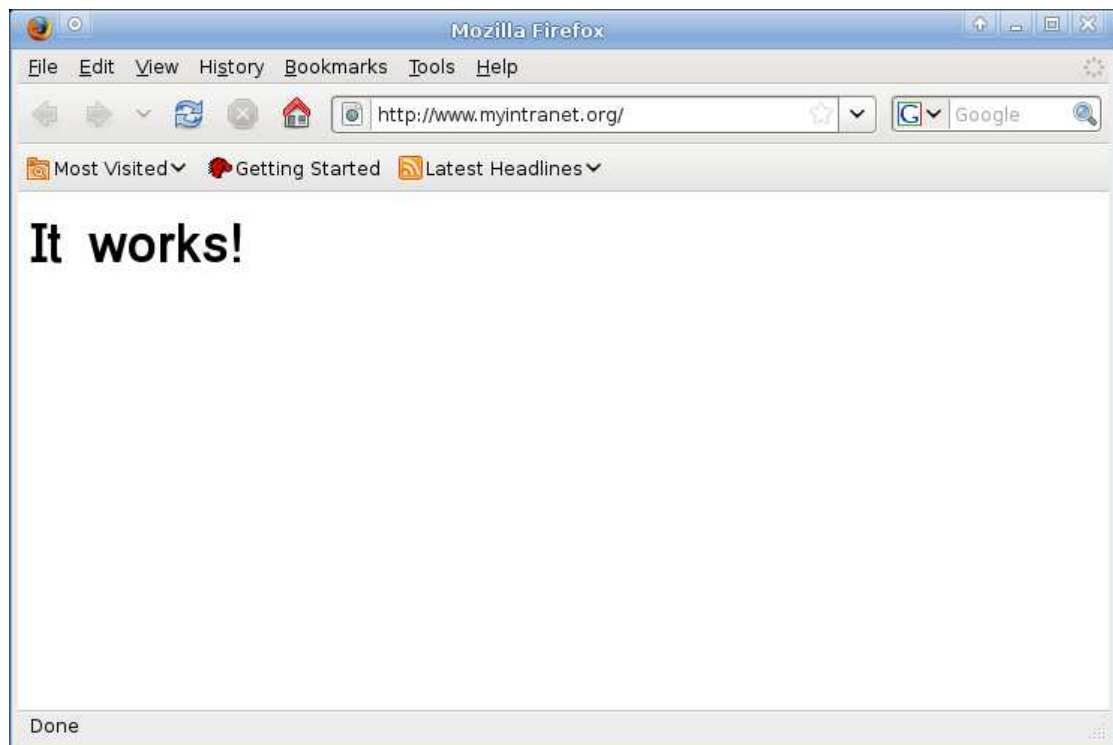
root@ubuntu-server:~#

ติดตั้งเสร็จก็รีสตาร์ท apache สักครั้งเพื่อความสำเร็จของการติดตั้ง ด้วยคำสั่ง

```
root@webserver:/var/www# /etc/init.d/apache2 restart
* Restarting web server apache2 [ OK ]
root@webserver:/
var/www#
```

ทดสอบการทำงานของ Apache2

ทดสอบการทำงานของเว็บเซิร์ฟเวอร์โดยการเปิดโปรแกรมเบราว์เซอร์ แล้วเรียก URL
www.myintranet.org ก็จะได้หน้าเว็บมีข้อความว่า It works!



ทดสอบการทำงานของ PHP

ทดสอบการทำงานของ PHP โดยการสร้างไฟล์ `phpinfo.php` แล้วเก็บไว้ที่ `/var/www` โดยมีข้อความภายในไฟล์ ต่อไปนี้

```
<?php  
phpinfo();  
?>
```

ทดสอบโดยการเรียก URL `http://www.myintranet.org/phpinfo.php`



ไดเรกทอรีและไฟล์ที่เกี่ยวข้อง

`/var/www` เรียกว่า RootDocument คือที่เก็บไฟล์ html, php และอื่นๆ เพื่อนำไปแสดงผลบนหน้าเว็บ

`/etc/apache2/apache2.conf` ไฟล์คอนฟิกของ Apache

`/etc/apache2/conf.d/charset` ไฟล์คอนฟิกของ Apache ใช้สำหรับเปลี่ยนค่าปกติของภาษาที่แสดงผลบน

Browser

`/etc/apache` ไดเรกทอรีที่เก็บไฟล์คอนฟิกอื่นๆ ของ Apache

`/etc/php5/apache2/php.ini` ไฟล์คอนฟิกของ PHP

Dynamic Host Configuration Protocol

Dynamic Host Configuration Protocol (DHCP) หรือ (**DHCPv6**) เป็นโปรโตคอลที่ใช้ในเครือข่ายคอมพิวเตอร์ที่ทำงานแบบแม่ข่าย-ลูกข่าย โดยเครื่องคอมพิวเตอร์ลูกข่ายจะทำการร้องขอข้อมูลที่จำเป็น ในการเข้าร่วมเครือข่ายจากแม่ข่าย ซึ่งข้อมูลเหล่านี้รวมถึงหมายเลขไอพีที่ใช้ภายในเครือข่าย ซึ่งคอมพิวเตอร์แม่ข่ายเป็นฝ่ายกำหนดให้กับลูกข่าย

DHCP ได้รับการยอมรับเป็นมาตรฐานในเดือนตุลาคม ค.ศ. 1993 โดยมาแทนมาตรฐานเก่า BOOTP รุ่นล่าสุดของ DHCP คือ DHCPv6 ซึ่งใช้กับโปรโตคอล IPv6 ได้รับการอนุมัติเมื่อกรกฎาคม ค.ศ. 2003

การจ่ายหมายเลขไอพี

เซิร์ฟเวอร์ DHCP มีวิธีการจ่ายหมายเลขไอพี 3 วิธีดังนี้:

1. กำหนดด้วยตนเอง ผู้บริหารระบบเครือข่ายเป็นผู้กำหนดหมายเลข ไอพีที่ต้องการใช้สำหรับเครื่องลูกข่าย โดยใช้วิธีเทียบกับหมายเลข MAC
2. แบบอัตโนมัติ เซิร์ฟเวอร์ DHCP จะจ่ายหมายเลขไอพีที่ว่างอยู่ให้กับลูกข่ายโดยอัตโนมัติ ซึ่งจะใช้หมายเลขไอพีช่วงที่ผู้บริหารระบบกำหนดให้ใช้ได้ ไอพีที่จ่ายจะถูกใช้อย่างถาวร
3. แบบไดนามิก วิธีนี้เป็นวิธีเดียวที่สามารถนำหมายเลขไอพีมาใช้ซ้ำได้ เมื่อคอมพิวเตอร์ถูกเปิดเครื่องและเริ่มทำงาน เครื่องลูกข่ายจะขอหมายเลขไอพีจากเซิร์ฟเวอร์โดยอัตโนมัติ วิธีนี้ต่างกับแบบอัตโนมัติตรงที่หมายเลขไอพีในการทำงานแต่ละครั้ง ไม่จำเป็นต้องเป็นเลขเดิม

การติดตั้ง DHCP

ตัวอย่างข้อกำหนดของระบบ dhcp server

ติดตั้งบน server หมายเลข IP คือ 192.168.2.1

ใช้งานบน network หมายเลข 192.168.2.0 mask 255.255.255.0

แจกหมายเลข IP ตั้งแต่ 192.168.2.0 ถึง 192.168.2.50

default gateway คือ 192.168.2.1

1. ติดตั้งด้วยคำสั่ง **sudo apt-get install dhcp3-server**
2. การตั้งค่าต่างๆ ให้จัดการที่เพิ่ม config file ชื่อ dhcpd.conf อยู่ที่ /etc/dhcp3/dhcpd.conf
สั่งให้ dhcpd ทำงานด้วยคำสั่ง **sudo /etc/init.d/dhcp3-server start**

ต้องแก้ไขค่า subnet ให้ถูกต้องดังตัวอย่างคือ

```
subnet 192.168.2.0 netmask 255.255.255.0 {
range 192.168.2.20 192.168.2.50;
option domain-name-servers 192.168.2.1;
option routers 192.168.2.1;
}
```

3. สั่งให้ dhcpd ทำงานใหม่ด้วยคำสั่ง **sudo /etc/init.d/dhcp3-server force-reload**

4. การกำหนดให้ dhcpd ปล่อยเลข ip แบบ dynamic ให้ปรับปรุงแก้ไขเพิ่ม dhcpd.conf ดังตัวอย่าง

```
subnet 192.168.2.0 netmask 255.255.255.0 {
range 192.168.2.0 192.168.2.50;
}
```

5. การกำหนดให้ dhcpd ปล่อยเลข ip แบบ fix หมายเลข ip เข้ากับ MAC ของ lan card ให้ปรับปรุงแก้ไขเพิ่ม dhcpd.conf ดังตัวอย่าง

```
host ip1684-26 {
hardware ethernet 00:01:03:18:BB:4E;
fixed-address 192.168.2.51;
### อนุญาตให้เฉพาะ lan card ที่ได้ fix หมายเลข ip เข้าใช้งานเท่านั้น
deny unknown-clients;
}
```

6. การจัดทำแบบให้บางส่วนต้อง fix และบางส่วนเป็น dynamic ให้ปรับปรุงแก้ไขเพิ่ม dhcpd.conf ดังตัวอย่าง

```
host ncd2 { hardware ethernet 0:c0:c3:88:2d:81; }
host ncd3 { hardware ethernet 0:c0:c3:00:14:11; }
subnet 192.168.2.0 netmask 255.255.255.0 {
option routers 192.168.2.1;
# Unknown clients get this pool.
pool {
option domain-name-servers bogus.example.com;
max-lease-time 300;
range 192.168.2.200 192.168.2.253;
allow unknown-clients;
}
# Known clients get this pool.
pool {
option domain-name-servers ns1.example.com,ns2.example.com;
max-lease-time 28800;
```

```

range 192.168.2.100 192.168.2.199;
deny unknown-clients;
}
}

```

7. การสั่งยกเลิกไม่ให้ dhcpd ทำงานอัตโนมัติทุกครั้งที่ยูทิลิตี้เครื่อง ใช้คำสั่งว่า **sudo update-rc.d -f**

dhcp3-server remove

หากต้องการให้ dhcpd กลับมาทำงานอัตโนมัติทุกครั้งที่ยูทิลิตี้เครื่อง ใช้คำสั่งว่า **sudo update-rc.d dhcp3-server default**

(อ้างอิงจาก <http://opensource.cc.psu.ac.th/>)

Proxy Server คืออะไร

Proxy Server คือบริการของแม่ข่ายที่จัดเก็บ content ของ website ต่างๆ ที่เครื่องลูกข่ายได้ไปเยี่ยมชมขึ้นมา เพื่อที่ว่าเวลาที่เครื่องลูกข่ายเครื่องอื่น ไปบ้าง จะได้ไม่ต้องออกไปเอาข้อมูลในเน็ตจริงๆ แต่มาเอาที่เครื่องแม่ข่ายของเราแทน “แล้วมันจะคืออะไรถ้าเอามาเก็บไว้ที่เครื่องแม่ข่ายเรา” ที่แน่ๆ ถ้าเครื่องลูกข่ายออกไป Website เดียวกันหลายๆ เครื่อง แล้วไม่มีระบบ Proxy ละก็ เครื่องลูกข่ายแต่ละเครื่องก็จะต้องออกไปยังอินเทอร์เน็ตเพื่อไปเอาข้อมูล ทำให้เปลือง bandwidth และล่าช้าได้ แต่ถ้ามี Proxy Server อยู่แล้ว เครื่องลูกข่ายเครื่องแรกเท่านั้นที่จะออกอินเทอร์เน็ตไปเพื่อให้ได้ข้อมูลมา แล้วข้อมูลนั้น ก็จะถูกเก็บไว้ใน Proxy Server ของเรา ดังนั้น เมื่อเครื่องลูกข่ายเครื่องอื่นๆ ไป Website เดียวกันกับเครื่องแรก ก็ไปเอาที่ Proxy Server แทน ประโยชน์ด้านอื่นๆ หละ ถ้าในเครือข่ายเรามี Proxy Server แล้วละก็ ตัว Proxy Server จะทำหน้าที่เป็นคนกำหนดการออกไปยังอินเทอร์เน็ตของเครื่องลูกข่าย (เครื่องแรก ออกไปเอาจากอินเทอร์เน็ตมาจริงๆ เครื่องอื่น ไม่ได้ออกจริง Proxy จัดให้) ดังนั้น เราจึงสามารถปรับแต่ง Proxy Server ให้ควบคุมการออกไปยังอินเทอร์เน็ตแบบต่างๆ เช่น บางเครื่องออกไปยังอินเทอร์เน็ตได้ หรือ user บางคนจะเล่นอินเทอร์เน็ตได้ (ต้อง login ก่อน จึงจะรู้ว่าเล่นอินเทอร์เน็ตได้หรือไม่) ห้ามไป Website ที่ไม่เหมาะสม และกำหนดความสำคัญก็ได้ เช่น พนักงานธรรมดา ถูกบีบให้ใช้ได้น้อย ผู้บริหาร ออกได้เยอะกว่า แถมไม่ Block อีกต่างหาก เป็นต้น

ติดตั้ง squid

1. ติดตั้ง squid ด้วยคำสั่ง **sudo apt-get install squid squid-common**

2. ให้แก้ไขเพิ่ม /etc/squid/squid.conf กำหนดค่าตัวอย่างดังนี้

```
acl localnet src 192.168.2.0/16 * ประมาณบรรทัด 609
```

```
http_access allow localnet * ประมาณบรรทัด 674
```

****หากต้องการใช้งาน Proxy เป็น port อื่นเช่น 8080 ก็เปลี่ยนให้เป็น http_port 8080**

```
http_port 8080 * ประมาณบรรทัด 1110
```

ปรับขนาดพื้นที่เก็บ Cache

```
cache_dir ufs /var/spool/squid 1024 16 256 * ประมาณบรรทัด 1952
```

Save แล้วสั่ง stop service

```
# /etc/init.d/squid stop
```

สร้างพื้นที่เก็บ Cache ใหม่

```
# squid -z
```

สั่ง start service

```
# /etc/init.d/squid start
```

3. สั่งทำงานใหม่ด้วยคำสั่ง `sudo /etc/init.d/squid restart`

หากมีการเปลี่ยนแปลงค่าใน `/etc/squid/squid.conf` แล้วไม่ต้องการ Restart service ใหม่ ให้สั่งทำงานใหม่ด้วยคำสั่งว่า `sudo /etc/init.d/squid force-reload`

ทดสอบผลงาน proxy server นี้ โดยตั้งค่าใน browser ให้ชี้ไปที่ proxy server 192.168.2.1 port 8080

การใช้งาน Proxy เป็น Transparent Proxy

แก้ไขเพิ่ม `/etc/squid/squid.conf` ให้เพิ่ม option transparent ต่อท้าย port ดังนี้

```
http_port 8080 transparent * ประมาณบรรทัด 1110
```

ตรวจสอบเพื่อกเกตที่วิ่งผ่านเครื่อง Server นี้ ว่ามีการเรียกใช้งาน เว็บ (Port 80) หรือไม่ ถ้ามีก็บังคับให้ใช้งานผ่าน Squid โดยใช้คำสั่ง iptables

ตัวอย่าง กรณี Server มี 1 lan card

```
#echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
#iptables -F
```

```
#iptables -t nat -F
```

```
#iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --toport 8080
```

ตัวอย่าง กรณี Server มี 2 lan card โดย eth0 เป็น Public IP และ eth1 เป็น Private IP

```
#echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
#iptables -F
```

```
#iptables -t nat -F
#iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --toport
8080
```

ตัวอย่าง กรณี Server มี 2 lan card และทำ masquerade ไว้โดย eth0 เป็น Public IP และ eth1 เป็น Private IP

```
#echo 1 > /proc/sys/net/ipv4/ip_forward
#iptables -F
#iptables -t nat -F
#iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
#iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --toport
8080
```

การใช้ *Secure Shell* ในการเข้าถึงระบบจากระยะไกล

What are SSH, SFTP and SCP?

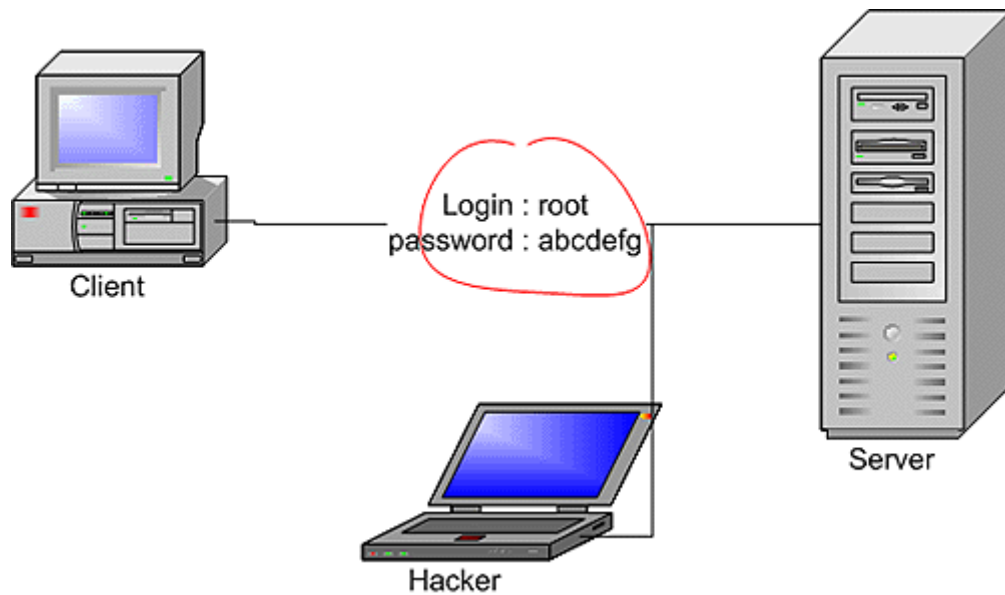
Secure Shell (SSH) เป็น โพรโทคอลในการสร้างการติดต่อเพื่อเข้าใช้งานระบบอย่างปลอดภัย มากกว่าการติดต่อแบบเดิมๆ ที่มีการส่งข้อมูลเป็นเพียงตัวอักษรเปล่าๆ (Plain text) โดยที่โพรโทคอลดังกล่าวจะทำการเข้ารหัสข้อมูลทุกอย่างไม่ว่าจะเป็น ชื่อผู้ใช้ รหัสผ่าน หรือข้อมูลอื่นๆ ก่อนที่จะทำการส่งไปยังเครื่องเซิร์ฟเวอร์ โดยปกตินิยมนำ SSH มาใช้งานแทน telnet เพราะมีความปลอดภัยมากกว่า

Secure File Transfer Protocol (SFTP) เป็น โพรโทคอลที่นำมาใช้แทน FTP โดยจะเป็นส่วนหนึ่งของ SSH ซึ่งจะมี sftpserv เป็น โปรแกรมที่รันอยู่ที่ฝั่งเซิร์ฟเวอร์ รอรับการติดต่อจากไคลเอนต์ผ่านทางคำสั่ง sftp บนระบบปฏิบัติการ linux และในระบบปฏิบัติการ Windows ก็มีโปรแกรมที่จะใช้สำหรับติดต่อและโอนถ่ายข้อมูลกับเซิร์ฟเวอร์ผ่าน SSH ด้วย เช่น SecureFX และ SSH Secure Shell Client เป็นต้น

Secure Copy (SCP) เป็นวิธีการหนึ่งในรับส่งข้อมูลข้ามระบบเครือข่าย โดยมีลักษณะเป็นการคัดลอกข้อมูลระหว่างเครื่อง 2 เครื่อง และอาศัยโพรโทคอล Secure Shell เช่นเดียวกับ SFTP ซึ่งจะแตกต่างกันที่ SFTP จะต้อง Login เข้าไปที่ระบบก่อนแล้วจึงสามารถที่จะรับส่งไฟล์ได้ แต่ SCP นั้นจะป้อนรหัสผ่านทุกครั้งที่ทำกรรับส่งไฟล์

Why Use SSH?

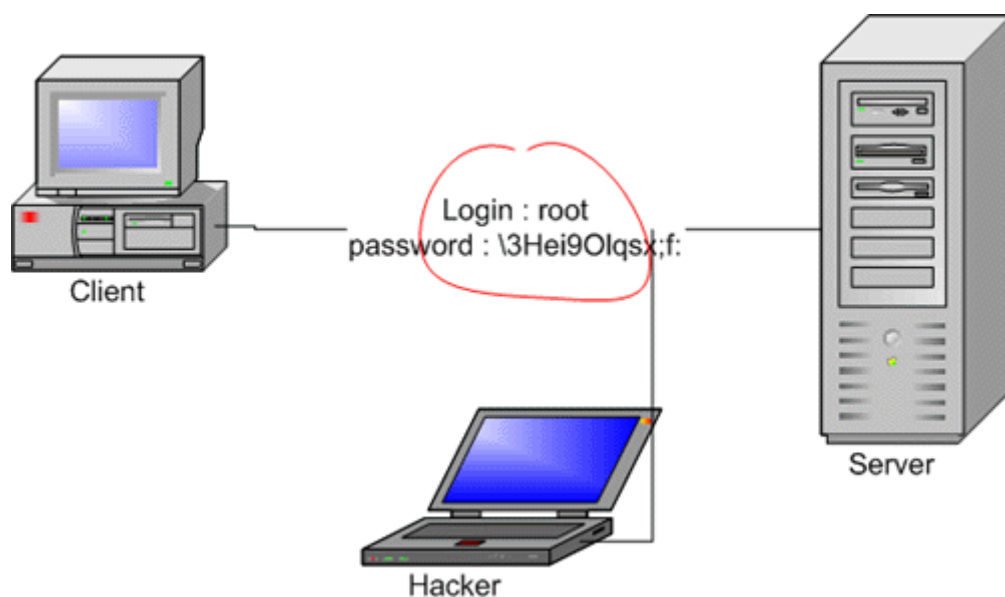
วิธีการส่วนมากที่ผู้โจมตีใช้กันคือการดักรับโมยข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์ผ่านทางโพรโทคอล telnet ซึ่งข้อมูลที่ดักรับได้ก็จะเป็นชื่อที่ใช้ Login และรหัสผ่าน ที่ไม่ได้ถูกเข้ารหัสไว้ โดยใช้โปรแกรมประเภท Sniffer ทั่วๆ ไป ดังรูป



รูป แสดงผลจากการใช้ telnet ในการทำ remote access ไปยังเซิร์ฟเวอร์และผู้โจมตีสามารถดักข้อมูลได้

และถ้าเกิดกรณีที่ผู้โจมตีดักรับ โมยชื่อผู้ใช้คือ root และรหัสผ่าน ความเสียหายที่ตามมาอาจจะมากจนเกินที่จะแก้ไขได้

ดังนั้นวิธีแก้ไขคือ ยกเลิกการใช้โปรโตคอล telnet โดยใช้โปรโตคอล SSH แทน และใช้ SFTP แทนโปรโตคอลของ FTP ทั้งนี้เพื่อความปลอดภัยจากการถูกดักข้อมูลขณะทำการส่ง ดังรูป

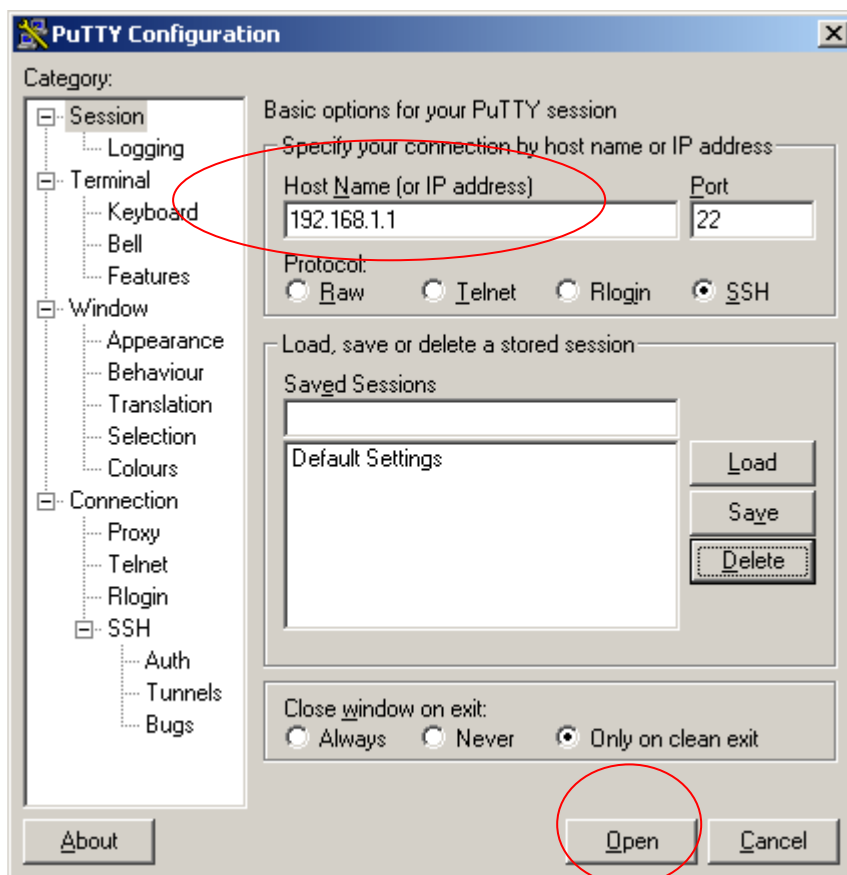


รูป แสดงผลจากการใช้ ssh จะสังเกตว่ารหัสผ่านที่ส่งจากไคลเอ็นต์นั้นจะถูกเข้ารหัสไว้

ถ้าใช้ระบบปฏิบัติการ Windows จะมีโปรแกรมที่ใช้ในการทำ Remote Login หลายโปรแกรม ยกตัวอย่างเช่น

PuTTY เป็นซอฟต์แวร์ฟรี ที่มีความสามารถในการทำ Remote Login และยังมีโปรแกรมที่อาศัย โพรโทคอล SSH ด้วยเช่น PSFTP (ใช้ในการรับส่งไฟล์ผ่าน Secure Shell) PSCP (ใช้ในการคัดลอกไฟล์ ผ่าน Secure Shell) เป็นต้น ซึ่งสามารถดาวน์โหลดได้ที่

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>



รูป แสดงหน้าแรกของโปรแกรม PuTTY