**A logo with a design on it

Description automatically generated**

**ใบงานที่ 2**

**เรื่อง Basic commands**

**เสนอ**

**อาจารย์ปิยพล ยืนยงสถาวร**

**จัดทำโดย**

**นายอธิศ สุนทโรดม 65543206086-2**

**ใบงานนี้เป็นส่วนหนึ่งของรายวิชา ระบบปฏิบัติการ**

**หลักสูตรวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์**

**มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา**

**ประจำภาคที่ 2 ปีการศึกษา 2566**

**ใบงานที่ 2**

**Basic commands**

**ขั้นตอนการทดลอง**

1. ให้สมัครสมาชิกเข้าใช้งานระบบใน <https://www.webminal.org/>
2. Login เข้าระบบ และเปิดเมนู Terminal
3. ปฏิบัติการทดลองคำสั่งใน Lessons 1 – 11
4. บันทึกการทดลอง และผลการทดลอง
5. อธิบายการปฏิบัติแต่ละขั้นตอนอย่างละเอียด พร้อมบันทึกภาพประกอบการปฏิบัติ
6. สรุปผลการทดลอง
7. จัดรูปแบบเอกสารใบงานให้เรียบร้อย
8. บันทึกไฟล์ในรูปแบบ PDF ส่งใน Assignment LAB2 : Basic commands

**บันทึกการทดลอง และผลการทดลอง**

Lesson1 - Basic commands to navigate directories

Lesson2 - Create files, display contents and stats

Lesson3 - Copy, rename, delete files

Lesson4 - Basic process commands

Lesson5 - Manipulate or parse file contents

Lesson6 - Changing file attributes

Lesson7 - Locate file and its type

Lesson8 - System and user details

Lesson9 - Linux Process Basic commands 1

Lesson10 - Linux Process Basic commands 2

Lesson11 - Linux Process states

**ขั้นตอนการทดลอง**

**Lesson1 - Basic commands to navigate directories**

A close-up of a number

Description automatically generated

A white background with black text

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A close up of a computer code

Description automatically generated

|  |  |
| --- | --- |
| **คำสั่ง** | **คำอธิบาย** |
| **pwd** | จะแสดงไดเร็กทอรีการทำงานปัจจุบันของคุณ ใช่แล้ว บ้านของคุณคือไดเร็กทอรี ตอนนี้ให้ลองสร้างไดเร็กทอรีใหม่พิมพ์ข้อความต่อไปนี้บนพรอมต์ |
| **mkdir dir1** | คือการสร้างโฟลเดอร์ชื่อ dir 1 |
| **ls** | ระบุ dir1 dir2 เป็นเนื้อหาไดเรกทอรีใช่ไหม นั่นคือสิ่งที่เราต้องการ |
| **ls -r** | ใช้เพื่อแสดงรายการไฟล์และไดเรกทอรีในลำดับย้อนกลับ (reverse order) หรือจากสุดท้ายขึ้นมาที่ต้นทาง นั่นคือแสดงไฟล์และไดเรกทอรีตามลำดับตรงกันข้ามกับที่ปกติ (จากท้ายสู่ต้น) ซึ่งช่วยให้คุณดูได้ง่ายว่าไฟล์ไหนถูกสร้างหรือแก้ไขล่าสุดก่อน โดยปกติ ls จะแสดงในลำดับปกติ (จากต้นทางไปสู่ท้าย) ถ้าเพิ่ม -r จะกลับลำดับนั้นให้เป็นจากท้ายสู่ต้น. |
| **cd dir2** | ซีดีคือการเข้าถึงไดเรกทอรี่ตามชื่อที่ตั้ง |
| **cd ..** | ถอยออกจากไดเรกทอรี่ |
| **cd** | จะย้ายไปยังโฮมไดเร็กตอรี่ของคุณ |

**Lesson2 - Create files, display contents and stats**

A screenshot of a computer screen

Description automatically generated

**รูปตัวอย่าง Lesson2**

|  |  |
| --- | --- |
| **คำสั่ง** | **คำอธิบาย** |
| **touch file1.txt** | สร้างไฟล์ใหม่ที่ชื่อ "file1.txt" ถ้าไฟล์นี้ยังไม่มีอยู่ คำสั่ง touch จะใช้สร้างไฟล์ใหม่ และหากไฟล์นี้มีอยู่แล้ว คำสั่ง touch จะใช้ในการอัปเดต timestamp (เวลาที่สร้างและแก้ไขล่าสุดของไฟล์) โดยไม่เปลี่ยนเนื้อหาของไฟล์. |
| **dir** | ใช้เพื่อแสดงรายการเนื้อหาไดเร็กทอรี ใช่แล้ว ตามที่คุณเดาถูกต้อง dir เทียบเท่ากับ ls -C -b (ฉันรู้ว่าคุณไม่ได้เดาว่า  นั่นคือ โดยค่าเริ่มต้น ไฟล์จะแสดงรายการในคอลัมน์ จัดเรียงในแนวตั้ง และอักขระพิเศษจะแสดงด้วยลำดับหลีกของแบ็กสแลช |
| **Clear** | เคลียหน้าเทอมินอล |
| **echo "hello" > hello.txt** | ใช้ในการสร้างไฟล์ชื่อ "hello.txt" และเขียนข้อความ "hello" เข้าไปในไฟล์นั้น โดยถ้าไฟล์ "hello.txt" มีอยู่แล้ว, คำสั่งนี้จะเขียนทับข้อมูลที่มีอยู่ในไฟล์ด้วยข้อความ "hello". |
| **cat hello.txt** | ใช้ในการแสดงเนื้อหาของไฟล์ "hello.txt" บนหน้าจอ. |
| **head -2 hello.txt** | ใช้ในการแสดงบางส่วนหรือ "หัว" (head) ของไฟล์ "hello.txt" บนหน้าจอ, โดยมีตัวเลข -2 ระบุว่าเราต้องการแสดงเพียง 2 บรรทัดแรกของไฟล์. |
| **tail hello.txt** | ใช้ในการแสดงบางส่วนหรือ "ท้าย" (tail) ของไฟล์ "hello.txt" บนหน้าจอ. คำสั่งนี้จะแสดงทั้งหมดของไฟล์ "hello.txt" หากไฟล์นี้มีขนาดไม่เกินจำนวนบรรทัดที่ระบุไว้ตามค่าตัวเลือก (default คือ 10 บรรทัด). |
| **stat hello.txt** | ใช้ในการแสดงข้อมูลสถิติ (statistical information) เกี่ยวกับไฟล์หรือไดเรกทอรีที่ระบุ. |
| **stat dir1** | จะแสดงข้อมูลที่เกี่ยวข้องกับไดเรกทอรีนั้น ๆ รวมถึงข้อมูลเกี่ยวกับการสิทธิ์, inode, วันที่และเวลา, และอื่น ๆ. |

**Lesson3 - Copy,rename,delete files**

A screenshot of a computer screen

Description automatically generated

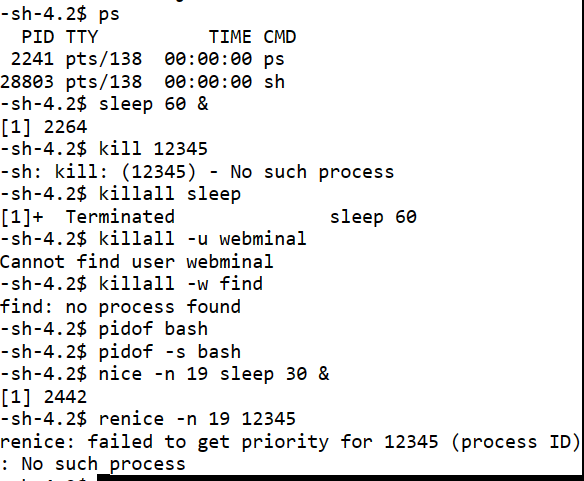
A computer screen shot of a computer code

Description automatically generated

**รูปตัวอย่าง Lesson3**

|  |  |
| --- | --- |
| **คำสั่ง** | **คำอธิบาย** |
| **du** | ใช้ในการแสดงขนาดที่ใช้พื้นที่บนดิสก์ของไดเรกทอรีและไฟล์. คำสั่งนี้สามารถใช้งานร่วมกับตัวเลือกต่าง ๆ เพื่อแสดงข้อมูลเพิ่มเติม เช่น ขนาดทั้งหมดที่ใช้, ขนาดไฟล์แต่ละไฟล์, หรือแสดงในรูปแบบที่เข้าใจง่าย. |
| **du -xh ~** | ใช้ในการแสดงขนาดที่ใช้พื้นที่บนดิสก์ของไดเรกทอรีที่ถูกระบุ (ในที่นี้คือ ~ ซึ่งแทนไดเรกทอรีของผู้ใช้ปัจจุบัน) โดยไม่สนใจการเข้าถึงไฟล์หรือไดเรกทอรีที่ตั้งอยู่ในไดเรกทอรีอื่น.  du: คำสั่งแสดงขนาดที่ใช้พื้นที่บนดิสก์ของไดเรกทอรีและไฟล์.  -x: กำหนดให้คำสั่งไม่ค้นหาขนาดที่ใช้พื้นที่บนดิสก์ของไดเรกทอรีและไฟล์ที่อยู่ในไดเรกทอรีที่ต่างจากไดเรกทอรี  -h: แสดงขนาดในรูปแบบที่อ่านง่าย (human-readable). |
| **cp -v hello.txt dir2** | ใช้ในการคัดลอก (copy) ไฟล์ "hello.txt" ไปยังไดเรกทอรี "dir2" และแสดงข้อความที่ระบุว่าไฟล์ถูกคัดลอก |
| **cp -vr dir2/\*.txt dir2/dir3** | ใช้ในการคัดลอกไฟล์ทั้งหมดที่มีนามสกุล ".txt" จากไดเรกทอรี "dir2" ไปยังไดเรกทอรี "dir3" และแสดงข้อความที่ระบุว่าไฟล์ถูกคัดลอก |
| **md5sum hello.txt** | ใช้ในการคำนวณและแสดงค่า MD5 checksum (hash) ของไฟล์. MD5 checksum เป็นตัวเลขที่เกิดจากการประมวลผลข้อมูลในไฟล์เพื่อตรวจสอบความถูกต้องของไฟล์นั้น. |
| **mv dir2/\*.txt dir5** | ใช้ในการย้าย (หรือเปลี่ยนชื่อ) ไฟล์ทั้งหมดที่มีนามสกุล ".txt" จากไดเรกทอรี "dir2" ไปยังไดเรกทอรี "dir5". |
| **ln dir2/dir3/dir4/hi.txt hello** | ใช้ในการสร้าง symbolic link (ที่ระบุโดยตัวชี้ไปยังไฟล์หรือไดเรกทอรีอื่น) ของไฟล์ "dir2/dir3/dir4/hi.txt" ในไดเรกทอรีปัจจุบัน และให้ symbolic link มีชื่อ "hello". |
| **ln -s dir2/dir3/dir4/hi.txt**  **softlink** | ใช้สร้าง symbolic link ที่ชื่อ "softlink" และชี้ไปยังไฟล์ "dir2/dir3/dir4/hi.txt".  ln: คำสั่งสร้าง symbolic link หรือ hard link.  -s: เป็นตัวเลือกที่ระบุให้ ln สร้าง symbolic link.  dir2/dir3/dir4/hi.txt: ไฟล์ที่เราต้องการสร้าง symbolic link ชี้ไปยัง.  softlink: ชื่อ symbolic link ที่เราต้องการสร้าง. |
| **rm -i file2.txt** | ใช้ในการลบไฟล์ "file2.txt" โดยให้ระบบถามยืนยันจากผู้ใช้ก่อนที่จะทำการลบ (interactive mode).  rm: คำสั่งลบไฟล์หรือไดเรกทอรี.  -i: เปิดโหมด interactive ซึ่งจะระบุให้ระบบถามคำยืนยันจากผู้ใช้ก่อนที่จะทำการลบ. |
| **rm -ri dir50/\*** | ใช้ในการลบทั้งหมดของไฟล์และไดเรกทอรีที่อยู่ในไดเรกทอรี "dir50" โดยให้ระบบถามคำยืนยันจากผู้ใช้ก่อนที่จะทำการลบแบบแอคชั่นตลอด. |
| **rm -rf junk/\*** | ใช้ในการลบทั้งหมดของไฟล์และไดเรกทอรีที่อยู่ในไดเรกทอรี "junk" โดยไม่ต้องถามคำยืนยันจากผู้ใช้ (force mode). |
| **rmdir dir50** | ใช้เพียงเท่านั้นสำหรับการลบไดเรกทอรีที่ว่างเปล่าเท่านั้น. |

**Lesson4 - Basic process commands**



A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**รูปตัวอย่าง Lesson4**

|  |  |
| --- | --- |
| **คำสั่ง** | **คำอธิบาย** |
| **Ps** | ใช้ในการแสดงข้อมูลเกี่ยวกับกระบวนการ (processes) ที่กำลังทำงานในระบบ. คำสั่งนี้ไม่แสดงกระบวนการของทุกๆ ผู้ใช้, แต่แสดงกระบวนการของผู้ใช้ปัจจุบัน. |
| **sleep 60 &** | ใช้ในการให้ระบบหยุดการทำงาน (sleep) เป็นเวลา 60 วินาที และใช้สัญลักษณ์ & เพื่อให้กระบวนการทำงานที่สร้างขึ้น (ในที่นี้คือ sleep) ทำงานในพื้นหลัง (background). |
| **kill 12345** | ใช้ในการส่งสัญญาณการสิ้นสุด (terminate signal) ไปยังกระบวนการที่มี Process ID (PID) เท่ากับ 12345. สัญญาณการสิ้นสุดทำให้กระบวนการหยุดการทำงานหากคำสั่งทำงานได้สำเร็จ, กระบวนการที่มี PID เท่ากับ 12345 จะถูกสิ้นสุด. หากไม่มีข้อผิดพลาด, kill จะไม่ส่งข้อความยืนยันกลับ. |
| **killall sleep** | ใช้ในการส่งสัญญาณการสิ้นสุดไปยังกระบวนการทั้งหมดที่ชื่อ "sleep". คำสั่งนี้มักใช้เพื่อหยุดการทำงานของกระบวนการทั้งหมดที่ชื่อ "sleep" ที่กำลังทำงานในระบบ. |
| **killall -u webminal** | ใช้ในการส่งสัญญาณการสิ้นสุดไปยังกระบวนการทั้งหมดที่ถูกเริ่มขึ้นโดยผู้ใช้ที่มีชื่อ "webminal". คำสั่งนี้จะหยุดการทำงานของกระบวนการทั้งหมดที่เป็นของผู้ใช้ "webminal". |
| **killall -w find** | ใช้ในการส่งสัญญาณการสิ้นสุดไปยังกระบวนการทั้งหมดที่มีชื่อ "find" และรอให้กระบวนการนั้นจบการทำงานก่อนที่จะสิ้นสุด. การใช้ตัวเลือก -w (wait) ทำให้คำสั่งรอให้กระบวนการ "find" ทุกรายการที่กำลังทำงานเสร็จสิ้นก่อนที่จะส่งสัญญาณการสิ้นสุด.  ในกรณีที่มีหลายกระบวนการ "find" ทำงานพร้อมกัน, -w จะทำให้ killall รอทุกรายการจนกว่ากระบวนการทั้งหมดจะเสร็จสิ้นก่อนที่จะทำการสิ้นสุด. ควรใช้ -w ในกรณีที่คุณต้องการให้การสิ้นสุดกระบวนการเสร็จสมบูรณ์ก่อนที่คำสั่ง killall จะถูกตั้งตาม. |
| **pidof bash** | ใช้ในการแสดง Process ID (PID) ของกระบวนการ (process) ที่ชื่อ "bash" |
| **pidof -s bash** | ใช้ในการแสดง Process ID (PID) ของกระบวนการ (process) ที่ชื่อ "bash" ที่กำลังทำงานในระบบปฏิบัติการ Linux และให้ผลลัพธ์เป็น PID เดียวที่ตรงกับกระบวนการแรกที่พบ. |
| **nice -n 19 sleep 30 &** | ใช้ในการเริ่มกระบวนการ sleep ที่มีความลำบาก (niceness) ที่สูงมาก (19) ในลักษณะ background ในระบบปฏิบัติการ Linux  nice: คำสั่งที่ใช้ในการเปลี่ยนความลำบากของกระบวนการ, ทำให้กระบวนการมีลำบากต่ำลงหรือสูงขึ้น.  -n 19: ตัวเลือก -n ใช้ในการกำหนดความลำบาก (niceness) ของกระบวนการ. ค่า niceness อยู่ในช่วง -20 (สูงสุด) ถึง 19 (ต่ำสุด). ค่า niceness สูงแสดงถึงการทำงานที่น้อยสุดของ CPU.  ในที่นี้, nice -n 19 ทำให้กระบวนการ sleep มีความลำบากสูงมาก (จำนวนความลำบากต่ำ) และ & ทำให้กระบวนการทำงานในพื้นหลัง.  กระบวนการ sleep 30 จะหยุดการทำงานเป็นเวลา 30 วินาที โดยไม่ใช้มีการใช้ทรัพยากร CPU มากนัก, เนื่องจากมีความลำบากสูง. |
| **renice -n 19 12345** | ใช้ในการเปลี่ยนความลำบาก (niceness) ของกระบวนการที่มี Process ID (PID) เท่ากับ 12345 ในระบบ |
| **top** | ใช้ในการแสดงข้อมูลเกี่ยวกับประสิทธิภาพของระบบและกระบวนการที่กำลังทำงานในระบบปฏิบัติการ Linux. เมื่อคุณป้อน top ในหน้าต่างคำสั่ง, มันจะแสดงตารางที่ประกอบด้วยข้อมูลต่าง ๆ เกี่ยวกับ CPU, หน่วยความจำ (Memory), และกระบวนการที่กำลังทำงาน. |
| **pstree -p** | ใช้ในการแสดงโครงสร้างของกระบวนการ (process tree) ในระบบปฏิบัติการ Linux พร้อมกับ Process ID (PID) ของแต่ละกระบวนการ. |
| **time ls -l** | ใช้ในการวัดเวลาที่ใช้ในการทำงานของคำสั่ง ls -l ซึ่งในที่นี้เป็นการแสดงรายการไฟล์และไดเรกทอรีในรูปแบบละเอียด  ผลลัพธ์ที่ได้จะแบ่งเป็นสามส่วน:  real: เวลาที่ใช้ทั้งหมดตั้งแต่เริ่มต้นคำสั่งจนถึงสิ้นสุด.  user: เวลาที่ CPU ใช้ในการประมวลผลคำสั่ง.  sys: เวลาที่ CPU ใช้ในการประมวลผลงานที่เกี่ยวข้องกับระบบ (system-related tasks). |

**Lesson5 - Manipulate or parse file contents**

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer program

Description automatically generated

**รูปตัวอย่าง Lesson5**

|  |  |
| --- | --- |
| **คำสั่ง** | **คำอธิบาย** |
| **grep "linux" hello** | ใช้ในการค้นหาข้อความ "linux" ในไฟล์ที่ชื่อ "hello"  grep: เป็นคำสั่งที่ใช้ในการค้นหาข้อความในไฟล์.  "linux": เป็นข้อความที่คุณกำลังค้นหา.  hello: เป็นชื่อของไฟล์ที่คุณต้องการค้นหาใน. |
| **grep -r 'Hello' .** | ใช้ในการค้นหาข้อความ "Hello" ในไฟล์ทั้งหมดในไดเรกทอรีปัจจุบัน (รวมถึงไดเรกทอรีย่อยทุกๆ ระดับ) ในระบบปฏิบัติการ Linux.  grep: เป็นคำสั่งที่ใช้ในการค้นหาข้อความในไฟล์.  -r: ตัวเลือกที่ใช้ในการค้นหาไฟล์ทั้งหมดในไดเรกทอรีและไดเรกทอรีย่อย.  'Hello': เป็นข้อความที่คุณกำลังค้นหา.  .: เป็นพาร์ทที่ระบุว่าคุณต้องการค้นหาในไดเรกทอรีปัจจุบัน. |
| **grep -i 'lINUX'** | ใช้ในการค้นหาข้อความ "lINUX" (โดยไม่สนใจตัวพิมพ์ใหญ่-เล็ก) ในไฟล์ที่ชื่อ "hello".  grep: เป็นคำสั่งที่ใช้ในการค้นหาข้อความในไฟล์.  -i: ตัวเลือกที่ใช้ในการทำให้ค้นหาเป็น case-insensitive, ซึ่งหมายความว่าไม่สนใจตัวพิมพ์ใหญ่-เล็ก. |
| **grep -n 'linux' hello** | ใช้ในการค้นหาข้อความ "linux" ในไฟล์ที่ชื่อ "hello" และแสดงหมายเลขบรรทัดที่พบข้อความนั้นด้วย.  grep: เป็นคำสั่งที่ใช้ในการค้นหาข้อความในไฟล์.  -n: ตัวเลือกที่ใช้ในการแสดงหมายเลขบรรทัดที่พบข้อความ. |
| **grep -v 'world' hello** | ใช้ในการแสดงบรรทัดที่ไม่มีข้อความ "world" ในไฟล์ที่ชื่อ "hello".  grep: เป็นคำสั่งที่ใช้ในการค้นหาข้อความในไฟล์.  -v: ตัวเลือกที่ใช้ในการแสดงบรรทัดที่ไม่ตรงกับข้อความที่กำลังค้นหา. |
| **wc -L hello** | ใช้ในการแสดงความยาวของบรรทัดที่ยาวที่สุดในไฟล์ที่ชื่อ "hello".  wc: เป็นคำสั่งที่ใช้ในการนับจำนวนบรรทัด, คำ, และตัวอักษรในไฟล์.  -L: ตัวเลือกที่ใช้ในการแสดงความยาวของบรรทัดที่ยาวที่สุด. |
| **cut -f1 -d' ' new.txt** | ใช้ในการแสดงเนื้อหาที่อยู่ในคอลัมน์ที่หนึ่ง (-f1) ของไฟล์ที่ชื่อ "new.txt" โดยใช้เครื่องหมายขีดว่างเป็นตัวคั่น (-d' ').  cut: เป็นคำสั่งที่ใช้ในการแสดงหรือตัดข้อมูลจากไฟล์.  -f1: ตัวเลือกที่กำหนดว่าควรแสดงคอลัมน์ที่เท่าไหร่. ในที่นี้คือคอลัมน์ที่หนึ่ง.  -d' ': ตัวเลือกที่กำหนดตัวคั่น (delimiter) ที่ใช้ในการแบ่งข้อมูล. ในที่นี้คือเครื่องหมายขีดว่าง. |
| **paste hello new.txt** | ใช้ในการนำเนื้อหาของไฟล์ "hello" และ "new.txt" มาต่อกันแนวตั้ง (vertically). |
| **sort new.txt** | ใช้ในการเรียงลำดับข้อมูลในไฟล์ "new.txt" ตามลำดับที่มีที่มีค่าน้อยไปที่มีค่ามาก. |
| **diff hello linux.txt** | ใช้ในการเปรียบเทียบเนื้อหาของไฟล์และแสดงการแตกต่างระหว่างไฟล์ทั้งสอง. |

**Lesson6 - Changing file attributes**



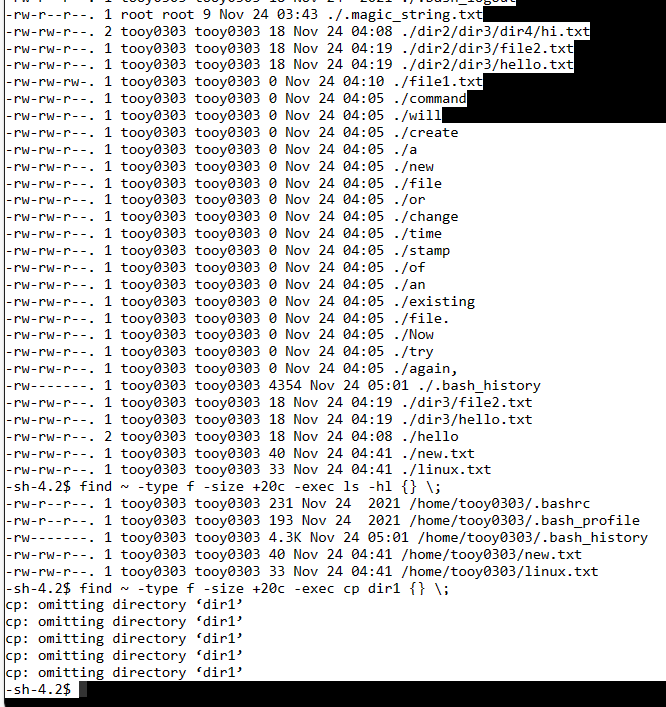
**รูปตัวอย่าง Lesson6**

|  |  |
| --- | --- |
| **คำสั่ง** | **คำอธิบาย** |
| **dirname dir2/dir3/dir4/hi.txt** | ใช้ในการแสดง directory ที่บรรจุ path ของไฟล์หรือ directory ที่ระบุ. |
| **basename dir2/dir3/dir4/hi.txt** | ใช้ในการแสดงชื่อของไฟล์หรือ directory ที่ถูกระบุใน path |
| **chmod -v 666 file1.txt** | ใช้ในการเปลี่ยนสิทธิ์การเข้าถึงของไฟล์ "file1.txt" เป็น 666 ซึ่งหมายถึงทุกคน (owner, group, others) สามารถอ่านและเขียนไฟล์นี้ได้.  chmod: เป็นคำสั่งที่ใช้ในการเปลี่ยนแปลงสิทธิ์การเข้าถึงของไฟล์หรือไดเรกทอรี.  -v: ตัวเลือกที่ใช้ในการแสดงผลการทำงานของคำสั่ง (verbose mode).  666: ค่าสิทธิ์ที่กำหนด 666 ให้ทุกคนมีสิทธิ์อ่านและเขียนไฟล์.  file1.txt: ชื่อของไฟล์ที่คุณต้องการเปลี่ยนแปลงสิทธิ์. |
| **chmod a+rw file1.txt** | ใช้ในการเปลี่ยนสิทธิ์การเข้าถึงของไฟล์ "file1.txt" ให้ทุกคน  (owner, group, others) มีสิทธิ์อ่านและเขียนไฟล์.  chmod: เป็นคำสั่งที่ใช้ในการเปลี่ยนแปลงสิทธิ์การเข้าถึงของไฟล์หรือไดเรกทอรี.  a: คือ "all" หรือทุกคน (owner, group, others).  +rw: เพิ่มสิทธิ์ให้ทุกคนอ่าน (read) และเขียน (write).  file1.txt: ชื่อของไฟล์ที่คุณต้องการเปลี่ยนแปลงสิทธิ์ |
| **chmod -R 644 ~/chmod\_dir** | ใช้ในการเปลี่ยนสิทธิ์การเข้าถึงของไดเรกทอรี "chmod\_dir" และไฟล์ทั้งหมดที่อยู่ในไดเรกทอรีนี้ให้มีสิทธิ์อ่าน (read) เฉพาะสำหรับเจ้าของไฟล์และสิทธิ์อ่าน (read) เฉพาะสำหรับกลุ่มและสิทธิ์ไม่มีสิทธิ์ในการเขียน (write) สำหรับทุกคนอื่น ๆ.  chmod: เป็นคำสั่งที่ใช้ในการเปลี่ยนแปลงสิทธิ์การเข้าถึงของไฟล์หรือไดเรกทอรี.  -R: ตัวเลือกที่ใช้ในการทำการเปลี่ยนแปลงทั้งหมดในโครงสร้างไดเรกทอรี (recursive).  644: ค่าสิทธิ์ที่กำหนด 644 ให้เจ้าของไฟล์มีสิทธิ์อ่าน (read) และเขียน (write) และกลุ่มและทุกคนมีสิทธิ์อ่าน (read) เท่านั้น.  ~/chmod\_dir: ที่อยู่ของไดเรกทอรีที่คุณต้องการเปลี่ยนแปลงสิทธิ์. |
| **chown root:staff -R ~/dir2** | ใช้ในการเปลี่ยนเจ้าของและกลุ่มของไดเรกทอรี "dir2" และไฟล์ทั้งหมดที่อยู่ใน "dir2" เป็น root เป็นเจ้าของ และ staff เป็นกลุ่ม.  chown: เป็นคำสั่งที่ใช้ในการเปลี่ยนเจ้าของและกลุ่มของไฟล์หรือไดเรกทอรี.  root:staff: ระบุเจ้าของและกลุ่มที่คุณต้องการกำหนด (root เป็นเจ้าของ, staff เป็นกลุ่ม).  -R: ตัวเลือกที่ใช้ในการทำการเปลี่ยนแปลงทั้งหมดในโครงสร้างไดเรกทอรี (recursive).  ~/dir2: ที่อยู่ของไดเรกทอรีที่คุณต้องการเปลี่ยนแปลงเจ้าของและกลุ่ม. |
| **chgrp root file1.txt** | ใช้ในการเปลี่ยนกลุ่มของไฟล์ "file1.txt" เป็นกลุ่ม "root".  chgrp: เป็นคำสั่งที่ใช้ในการเปลี่ยนกลุ่มของไฟล์หรือไดเรกทอรี.  root: ระบุกลุ่มที่คุณต้องการกำหนดให้กับไฟล์ (ในที่นี้คือ "root").  file1.txt: ชื่อของไฟล์ที่คุณต้องการเปลี่ยนกลุ่ม. |
| **chgrp -hR root dir2** | ใช้ในการเปลี่ยนกลุ่มของไดเรกทอรี "dir2" และไฟล์ทั้งหมดที่อยู่ใน "dir2" และซับไดเรกทอรีทั้งหมด เป็นกลุ่ม "root" โดยรวมกับตัวเลือก -R ทำให้มีการเปลี่ยนแปลงทั้งหมดในโครงสร้างไดเรกทอรี (recursive).  chgrp: เป็นคำสั่งที่ใช้ในการเปลี่ยนกลุ่มของไฟล์หรือไดเรกทอรี.  -h: ตัวเลือกที่ใช้ในกรณีที่กำหนดกลุ่มบน symbolic link ไฟล์หรือไดเรกทอรี จะเปลี่ยนกลุ่มของ symbolic link ไม่ได้ แต่จะเปลี่ยนกลุ่มของไฟล์หรือไดเรกทอรีที่ symbolic link ชี้ไป.  -R: ตัวเลือกที่ใช้ในการทำการเปลี่ยนแปลงทั้งหมดในโครงสร้างไดเรกทอรี (recursive).  root: ระบุกลุ่มที่คุณต้องการกำหนดให้กับไฟล์และไดเรกทอรี (ในที่นี้คือ "root").  dir2: ที่อยู่ของไดเรกทอรีที่คุณต้องการเปลี่ยนกลุ่ม. |

**Lesson7 - Locate file and its type**

A screen shot of a computer screen

Description automatically generated



**รูปตัวอย่าง Lesson7**

|  |  |
| --- | --- |
| **คำสั่ง** | **คำอธิบาย** |
| **file linux.txt** | ใช้ในการแสดงประเภทของไฟล์ "linux.txt" โดยตรวจสอบความหลากหลายของไฟล์ตามลายลักษณ์บนไฟล์. |
| **file /dev/null** | ใช้ในการแสดงประเภทของไฟล์ที่ตำแหน่ง "/dev/null". |
| **whereis ls** | ใช้ในการค้นหาตำแหน่งของไบนารี (binary), ไฟล์ความช่วยเหลือ (man page), และไฟล์ที่ทำให้โปรแกรมรันได้ (source code) ของคำสั่ง "ls" ในระบบของคุณ. |
| **whereis stdio.h** | ใช้ในการค้นหาตำแหน่งของไฟล์ "stdio.h" ในระบบของคุณ. |
| **php** | ใช้เรียกใช้ PHP interpreter ในโปรแกรมคอมพิวเตอร์หรือ command line interface (CLI). การเรียกใช้คำสั่ง php โดยอยู่ในโหมดแบบโต้ตอบ (interactive mode) จะเปิดหน้าต่างสำหรับป้อนคำสั่ง PHP ต่อมา. |
| **which php** | ใช้ในการแสดงตำแหน่งของไฟล์ทำงาน (executable) สำหรับโปรแกรม PHP ในระบบของคุณ.  **find ~ -name "linux.txt"** คือ ใช้ในการค้นหาไฟล์ที่มีชื่อ "linux.txt" ในโฮมไดเรกทอรีของผู้ใช้ปัจจุบันและตำแหน่งย่อยทั้งหมด.  ~: แทนโฮมไดเรกทอรีของผู้ใช้ปัจจุบัน.  -name "linux.txt": กำหนดเงื่อนไขในการค้นหา ในที่นี้คือ การค้นหาไฟล์ที่มีชื่อเป็น "linux.txt" |

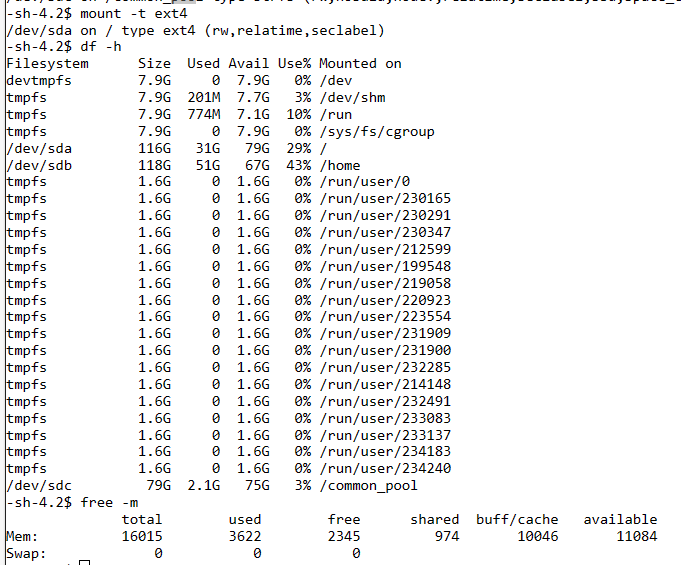
**Lesson8 - System and user details**

A screen shot of a computer

Description automatically generated

A black and white text

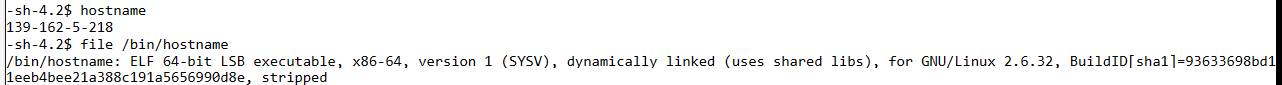
Description automatically generated



**รูปตัวอย่าง Lesson8**

|  |  |
| --- | --- |
| **คำสั่ง** | **คำอธิบาย** |
| **uptime** | ใช้เพื่อแสดงข้อมูลเกี่ยวกับระยะเวลาที่ระบบได้ทำงาน, จำนวนผู้ใช้ที่เข้าสู่ระบบ, และโหลดของระบบ (system load). |
| **date** | ใช้เพื่อแสดงเวลาและวันที่ปัจจุบันของระบบ. |
| **who -a** | ใช้ในการแสดงข้อมูลเกี่ยวกับผู้ใช้ที่เข้าสู่ระบบ รวมถึงข้อมูลเพิ่มเติมเกี่ยวกับการเชื่อมต่อ (login sessions). |
| **w** | ใช้ในการแสดงข้อมูลเกี่ยวกับผู้ใช้ที่เข้าสู่ระบบและกำลังทำงานในระบบ รวมถึงข้อมูลที่เกี่ยวข้องกับโหลดของระบบ (system load). |
| **mount** | ใช้ในการแสดงรายการของพาร์ติชันที่มีการติดตั้งและใช้งานในระบบไฟล์ของ Linux. ตำแหน่งการติดตั้ง, พาร์ทิชัน, และประเภทของระบบไฟล์ที่กำลังใช้งานจะถูกแสดงในผลลัพธ์. |
| **mount -t ext4** | ใช้ในการแสดงรายการของพาร์ติชันที่ใช้ระบบไฟล์ประเภท ext4 ที่ถูกติดตั้งและใช้งานในระบบ Linux. |
| **df -h** | ใช้ในการแสดงข้อมูลเกี่ยวกับการใช้พื้นที่ดิสก์ (disk space) ในระบบไฟล์. ตัวอักษร -h ในที่นี้หมายถึง "human-readable," ทำให้ข้อมูลแสดงในรูปแบบที่อ่านง่ายสำหรับมนุษย์. |
| **free -m** | ใช้ในการแสดงข้อมูลเกี่ยวกับการใช้งานหน่วยความจำ (RAM) ในระบบ. ตัวอักษร -m หมายถึง "megabytes," ทำให้ข้อมูลแสดงในหน่วยเป็นเมกะไบต์. |

**Lesson9 - Linux Process Basic commands 1**





A screenshot of a computer

Description automatically generated

**รูปตัวอย่าง Lesson9**

|  |  |
| --- | --- |
| **คำสั่ง** | **คำอธิบาย** |
| **hostname** | ใช้ในการแสดงชื่อของเครื่อง (hostname) ที่กำลังใช้งาน |
| **file /bin/hostname** | ใช้ในการแสดงประเภทของไฟล์ที่ตำแหน่ง "/bin/hostname" |
| **cat /bin/hostname** | เรามีอักขระที่อ่านได้และอ่านไม่ได้รวมกัน ที่ใดที่หนึ่งภายในเอาต์พุตเหล่านั้น เรากำลังบอกให้ Kernel อ่านชื่อไฟล์ '/etc/hosts' และค้นหารายการ '127.0.0.1' (ซึ่งโดยทั่วไปจะชี้ไปที่ชื่อเครื่องโฮสต์) และแสดงเนื้อหานั้น  เมื่อเราพิมพ์ไฟล์ปฏิบัติการ '/bin/hostname' ใน bash prompt เนื้อหาใดก็ตามที่คุณเห็นสองสามบรรทัดด้านบนจะถูกโหลดลงใน RAM เคอร์เนลปฏิบัติตามคำสั่งเหล่านั้นและดำเนินการ  สุดท้ายนี้ ต่อไปนี้เป็นคำจำกัดความของกระบวนกา  'กระบวนการไม่มีอะไรนอกจากเนื้อหาไฟล์ที่อยู่ใน RAM' |
| **ps** | ใช้ในการแสดงข้อมูลเกี่ยวกับกระบวนการ (processes) ที่กำลังทำงานในระบบ. |

**Lesson10 - Linux Process Basic commands 2**

A screenshot of a computer program

Description automatically generated

**รูปตัวอย่าง Lesson10**

เรารู้ว่ามีกระบวนการแรกที่ชื่อว่า 'init' พร้อมด้วย pid นี่คือพาเรนต์ของกระบวนการทั้งหมดในระบบ และกระบวนการชื่อ 'bash' โต้ตอบกับเคอร์เนลในนามของคำขอหรือคำสั่งของผู้ใช้

ตอนนี้เมื่อฉันเข้าสู่ระบบและพิมพ์ ps ทุกครั้งที่คุณเข้าสู่ระบบ ระบบจะสร้าง parent bash ใหม่ ในกรณีที่รหัสงาน Manager Bash ของเราคือ 5254 คุณรู้ไหมว่าแต่ละคำสั่งนั้นเป็นกระบวนการใช่ไหม? เรามาสร้างกระบวนการเล็กๆ น้อยๆ สำหรับเซสชันนี้กัน

**Background process**

พิม **sleep 5** มันค้างเป็นเวลา 5 วินาทีแล้วแจ้ง bash prompt ให้คุณอีกครั้ง

วิธีที่ bash ของเราสร้างกระบวนการลูกกำลังบล็อกการโทร หมายความว่ารันกระบวนการย่อยและรอให้เสร็จสิ้นแล้วกลับมาหาฉันอีกครั้ง กระบวนการเบื้องหลัง ดังนั้นเมื่อเรารันกระบวนการลูก (สลีป) เชลล์ 'Bash' จะรอให้มันเสร็จสิ้น

เราสามารถวิ่ง sleep 2 โดยไม่ต้องห้อยเป็นเวลา 5 วินาทีได้! และหมายเลขที่คุณเห็น 5781 คือรหัสกระบวนการลูกพื้นหลัง มันเป็นสำหรับการอ้างอิงของเรา จะตรวจสอบได้อย่างไรว่าเป็น pid ของเด็กจริงๆ? ง่าย ๆ เพียงเริ่มกระบวนการลูกในพื้นหลังและดำเนินการคำสั่ง 'ps'

sleep 5 pid ของเราตรงกับเอาต์พุต ps เรามีวิธีที่ดีกว่าในการแสดงภาพนี้ไหม? ใช่ทำ เรามีคำสั่งชื่อ 'pstree' ซึ่งจะบอกคุณถึงการแมประหว่างกระบวนการลูกและกระบวนการหลัก

มันบอกว่าเรามี Bash และโพรเซสลูกหนึ่งตัวชื่อ 'pstree' คอยดูแลลูกของเรา

ตอนนี้มันบอกว่า 'Bash' มีลูกสองคนชื่อ pstree และกระบวนการเบื้องหลัง 'sleep' pstree มีตัวเลือกให้พูดถึง -p ซึ่งแสดง pid ถัดจากชื่อกระบวนการ

pid ที่เราได้รับขณะเริ่มกระบวนการพื้นหลังเหมือนกับที่เราได้รับจากเอาต์พุต pstree

สมมติว่าเราได้เริ่มต้นงานเบื้องหลังที่ทำงานระยะยาว 4 งาน แต่ละงานใช้เวลา 145 วินาที และ 1 กระบวนการที่ยาวมากเป็นเวลา 3000 วินาที

**List background jobs**

ptree ให้ข้อมูลเกี่ยวกับงานทั้งหมด เราไม่ต้องการ pstree(6410) เพราะสนใจเฉพาะงานพื้นหลังเท่านั้น วิธีดูเฉพาะงานเหล่านั้น เพื่อจุดประสงค์นี้ เรามีคำสั่ง 'jobs' ที่จะให้ผลลัพธ์เช่น

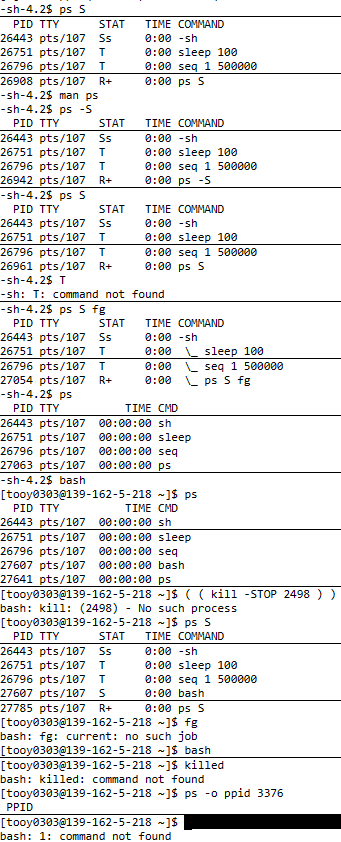
**Foreground process**

กระบวนการของเรา-[5] ทำงานเป็นเวลา 3,000 วินาที ซึ่งใช้เวลานานจึงจะเสร็จสมบูรณ์ พื้นหลังจะใช้เวลา CPU น้อยกว่าเมื่อเทียบกับกระบวนการที่ไม่ใช่พื้นหลัง กระบวนการเบื้องหน้า เรามาเข้าสู่กระบวนการเบื้องหน้ากันดีกว่า พิมพ์ fg ในกรณีของเรา เราต้องนำงานพื้นหลัง-5 มาด้วย

**Switch between foreground to background**

ไม่ต้องกังวล Linux มีความยืดหยุ่นมาก เราก็สามารถทำได้เช่นกัน เพียงกด 'ctrl+z' คุณจะได้รับเอาต์พุตแจ้งว่างานหยุดแล้ว

**Lesson11 - Linux Process states**



**รูปตัวอย่าง Lesson10**

|  |  |
| --- | --- |
| **คำสั่ง** | **คำอธิบาย** |
| **D** | การนอนหลับอย่างต่อเนื่อง (โดยปกติคือ IO) |
| **R** | กำลังรันหรือรันได้ (บนคิวรัน) |
| **S** | การนอนหลับแบบขัดจังหวะ (รอให้เหตุการณ์เสร็จสิ้น) |
| **T** | หยุดทำงานโดยสัญญาณควบคุมงาน |
| **X** | ตายแล้ว (ไม่ควรเห็น) |
| **ps -S** | STAT S ย่อมาจาก Interruptible sleep ที่นี่ bash/shell ของเรากำลังรอให้ลูก 28682 ดำเนินการให้เสร็จสิ้น 28682 อยู่ในสถานะกำลังทำงาน s - ย่อมาจาก session leader และ + บอกว่ามันจะทำงานในเบื้องหน้า |
| **ps S** | PID (Process ID): เลขระบุกระบวนการ ซึ่งเป็นตัวเลขที่ไม่ซ้ำกันที่ใช้ระบุแต่ละกระบวนการ.  TTY (Terminal): บอกถึงที่สัญญาณเอาต์พุตของกระบวนการ หรือ terminal ที่กระบวนการนั้นๆ  กำลังทำงานอยู่.  STAT (Status): แสดงสถานะปัจจุบันของกระบวนการ เช่น R (Running), S (Sleeping), Z (Zombie)ฯลฯ  TIME: เวลาที่ CPU ใช้ในการประมวลผลกระบวนการนั้นๆ.  COMMAND: คำสั่งที่ใช้เริ่มต้นกระบวนการนั้นๆ. |
| **seq 1 500000** | ใช้ในการสร้างลำดับตัวเลขที่เริ่มต้นที่ 1 และสิ้นสุดที่ 500,000 แล้วแสดงผลลัพธ์บนหน้าจอ. โดยปกติ, ผลลัพธ์จะมีจำนวนมากเพื่อแสดงลำดับตัวเลขทั้งหมด. |
| **Zombie process** | Zombie เป็นกระบวนการที่ถูกยกเลิกแต่ไม่ได้ถูกเก็บเกี่ยวโดยผู้ปกครอง เมื่อเราบอกว่าพ่อแม่ไม่ได้เก็บเกี่ยว เราหมายถึง 'ผู้ปกครองยังไม่ได้รับสถานะการออกจากลูก' ลูกดำเนินการเสร็จสิ้นพร้อมทั้งสถานะออกและรอให้ผู้ปกครองขอ นี่เป็นกรณีที่ยุ่งยากในการทำซ้ำ ให้ลอง: เรามีผู้นำเซสชันของเรา 2249 |
| **bash** | ชื่อของ Shell (Bourne Again SHell) ซึ่งเป็นโปรแกรมที่ให้สามารถใช้ในการประมวลผลคำสั่งที่คุณป้อนในระบบ Unix-like. ถ้าคุณพิมพ์ "bash" และกด Enter, นั้นจะทำให้คุณเข้าสู่โหมด Bash Shell ที่ให้คุณป้อนคำสั่งโดยตรง. คุณจะเห็นพรอมประสบการณ์สั่งคำสั่งที่สามารถใช้งานได้ใน Bash  พิม ps |

**Orphaned process**

เด็กกำพร้ากำลังดำเนินการโดยไม่มีผู้ปกครอง บิดามารดาเสียชีวิตโดยไม่ได้เก็บเกี่ยวบุตร (ได้รับสถานะออกจากบุตร) กระบวนการลูกที่ไม่มีพาเรนต์เหล่านี้เรียกว่ากระบวนการกำพร้า กระบวนการเหล่านี้ถูกนำมาใช้โดย init ซึ่งมี pid 1 มาลองสร้างกระบวนการที่ถูกละเลยซึ่งคล้ายกับขั้นตอนข้างต้น:

|  |  |
| --- | --- |
| **คำสั่ง** | **คำอธิบาย** |
| **( sleep 100 & ( kill -9 3329 ))** | ให้เรามาแยกแจงทีละส่วน:  sleep 100 &: คำสั่ง sleep นี้จะทำให้กระบวนการหยุดการทำงาน (sleep) นาน 100 วินาที และ & ทำให้กระบวนการนี้ทำงานใน background.  ( kill -9 3329 ): คำสั่ง kill นี้จะใช้สั่งการจบการทำงานของกระบวนการที่มี PID คือ 3329 โดยใช้สัญลักษณ์ -9 ซึ่งหมายถึงใช้สั่งการของการทำลาย (forceful kill). การใช้ () รวมคำสั่ง kill ใน subshell นี้ทำให้ทั้งคำสั่งทำงานในกระบวนการย่อยแยกต่างหาก.  การรวมทั้งหมดเข้าด้วยกัน, กระบวนการ sleep 100 จะทำงานใน background นาน 100 วินาที และพร้อมกับนั้น คำสั่ง kill -9 3329 จะถูกเรียกใช้เพื่อทำลายกระบวนการที่มี PID เท่ากับ 3329 โดยใช้การทำลายแบบเร่งด่วน |
| ps S | จะเห็นได้ว่า PID 3329 ถูก kill แล้ว |

**สรุปผลการทดลอง**

การฝึกใช้คำสั่งสำหรับการใช้งานเพื่อให้รู้ถึงการใช้งานของคำสั่งต่างๆ และให้สามารถนำไปใช้งานได้อย่างมีประสิธภาพ อย่างเช่นการใช้คำสั่งในการสร้างไฟล์ โดยการสร้างแบบหลายๆ โฟลเดอร์ mkdir หรือการสร้างไฟล์แบบเลือกโฟดเดอร์ทำให้เราไม่จำเป็นต้อง cd เข้าไปสร้างไฟล์แบบปกติจึงทำให้ช่วยประหยัดเวลาในการทำงานากยิ่งขึ้น