บทที่ 3

การออกแบบโปรแกรม

ในบทนี้กล่าวถึงการออกแบบโปรแกรมให้บริการถ่ายโอนแฟ้มให้สามารถรองรับการใช้ระบบรหัสผ่านแบบใช้ครั้งเดียว โดยการปรับปรุงโปรแกรม wu-ftpd รุ่น 2.4.2 beta 17 ในส่วนของการพิสูจน์ตัวจริง (Authentication) ได้แก่

-ฟังก์ชันที่เกี่ยวกับการดึงข้อมูลของผู้ใช้

-กระบวนงานที่ใช้ในการตรวจสอบรหัสผ่าน

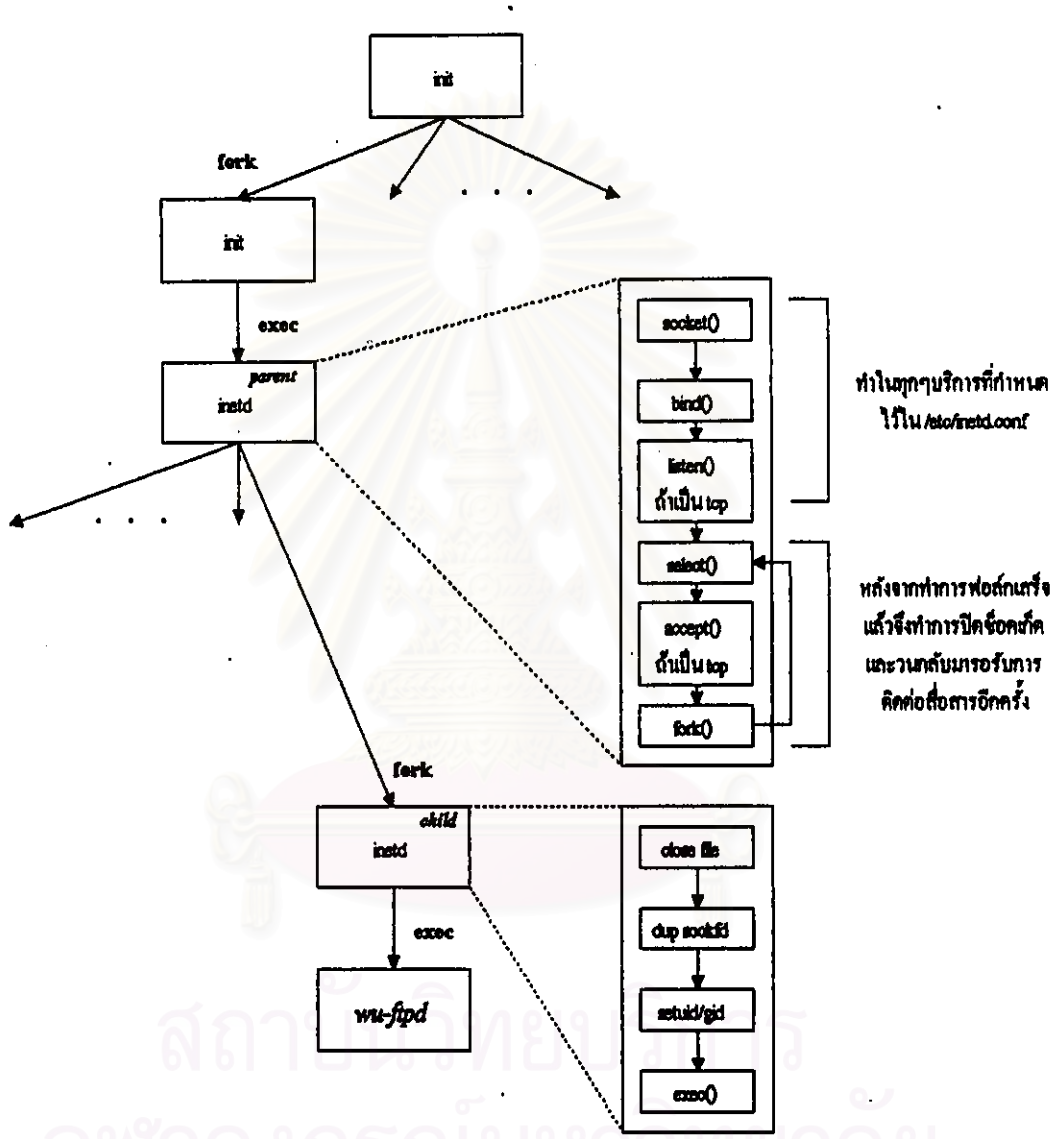
-กระบวนงานที่ใช้ในการติดต่อกับโปรแกรมตรวจสอบรหัสผ่านแบบใช้ครั้งเดียว

3.1 การทำงานของโปรแกรม wu-ftpd รุ่น 2.4.2 beta 17

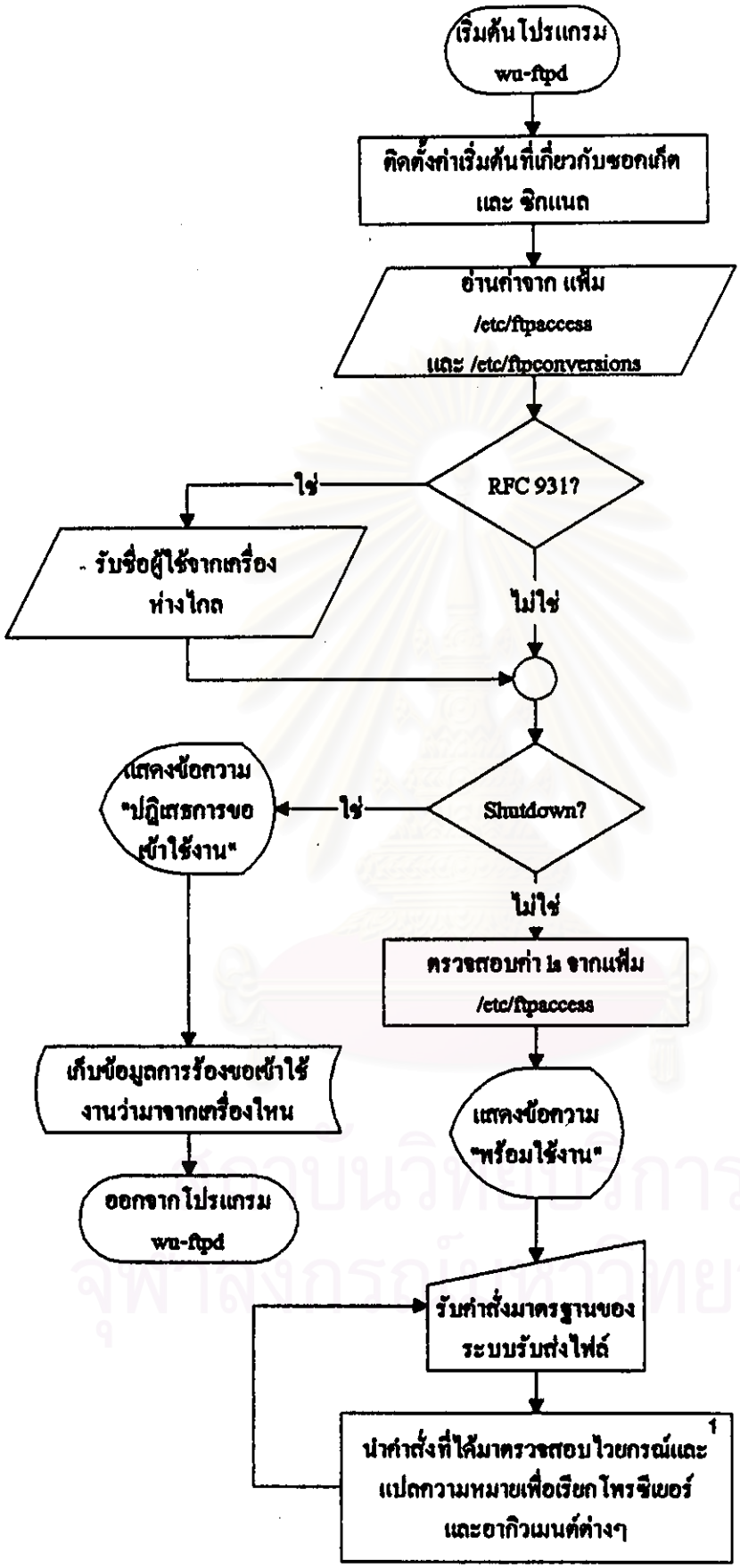
เมื่อระบบยูนิกซ์เริ่มทำงาน ใจกลาง (kernel) จะทำการเรียกโปรแกรม /etc/init ทำให้เกิดกระบวนการ init และ กระบวนการ init เรียกใช้งานแฟ้มข้อมูล /etc/rc (หากเป็นระบบยูนิกซ์ตระกูล System V จะมีการเรียกใช้งานแฟ้มข้อมูล /etc/inittab ก่อน) เพื่อเริ่มการทำงานกระบวนการดีมอนต่างๆ (daemon process) ซึ่งในระบบยูนิกซ์สามารถมีดีมอนต่างๆได้มากมาย ดีมอนที่ให้บริการโดยใช้ระบบเครือข่ายทีซีพี/ไอพีนั้นมีอยู่เป็นจำนวนมาก เช่น ดีมอนของเทลเน็ต ดีมอนของการถ่ายโอนแฟ้ม ดีมอนสำหรับลงบันทึกเข้าระยะไกล (remote login daemon) และอื่นๆ โดยที่แต่ละ กระบวนการต่างๆเหล่านี้จะทำงานใกล้เคียงกันมากในตอนต้น เช่น ต้องมีการสร้างซ็อกเก็ต (socket) ยึดเหนี่ยว (bind) แล้วจึงทำการรอการติดต่อสื่อสารจากช่องทาง (port) สื่อสาร เมื่อได้รับการติดต่อสื่อสารแล้วจึงทำการ ฟอร์ก (fork) กระบวนการให้กระบวนการลูก (child process) ทำงานแทน ส่วนกระบวนการพ่อ (parent process) จะทำการรอรับการติดต่อของผู้ใช้รายต่อไป ทำให้มีกระบวนการต่างๆมากมายในระบบยูนิกซ์

จึงมีการนำโปรแกรม inetd เข้ามาใช้งานโดยมีจะทำงานแทนงานซ้ำๆเหล่านั้น เพื่อให้มีกระบวนการในระบบน้อยกว่า รวมถึงการพัฒนาโปรแกรมดีมอนยังสามารถทำได้ง่ายโดยที่ไม่

ต้องมีการทำคำสั่งเริ่มต้นดังที่กล่าวมาแล้วข้างต้น แล้วจึงทำการ exec แดมอนต่างๆรวมถึงโปรแกรม wu-ftpd ด้วยดังรูปที่ 3.1

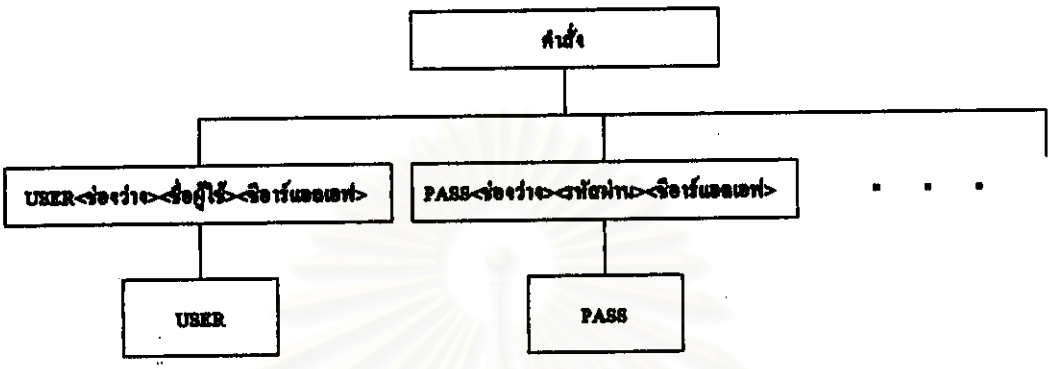


รูปที่ 3.1 แสดงลำดับการทำงานของโปรแกรม wu-ftpd โดยผ่านโปรแกรม inetd



รูปที่ 3.2 แสดงการทำงานของโปรแกรม wu-ftpd 2.4.2 beta 17

3.1.1 การนำคำสั่งมาแปลความหมายเพื่อเรียกกระบวนงาน(procedure)และอากิวเมนต์ (argument) ต่างๆนั้น คำสั่งจะเป็นคำสั่งที่มีอยู่ในอาร์เอฟซี 959



รูปที่ 3.3 แสดงการเรียกกระบวนงานจากคำสั่งที่ได้รับ

ดังรูปที่ 3.3 หาก คำสั่งที่ได้รับคือ

USER <ช่องว่าง> <ชื่อผู้ใช้> <ซีอาร์แอลเอฟ>

โปรแกรมจะทำการเรียก กระบวนงาน USER โดยให้ชื่อผู้ใช้เป็นอากิวเมนต์ หรือ อีกคำสั่งหนึ่งคือ

PASS <ช่องว่าง> <รหัสผ่าน><ซีอาร์แอลเอฟ>

โปรแกรมจะทำการเรียก กระบวนงาน PASS โดยให้รหัสผ่านเป็นอากิวเมนต์

คำสั่งที่เกี่ยวข้องกับการพิสูจน์ตัวจริง (authenticate) ได้แก่ คำสั่ง USER และ คำสั่ง PASS โดยคำสั่ง USER ทำหน้าที่ในการดึงข้อมูลผู้ใช้ก่อนการตรวจสอบรหัสผ่านด้วยคำสั่ง PASS ดังนั้นการปรับปรุงให้โปรแกรม wu-ftpd สามารถใช้งานระบบรหัสผ่านแบบใช้ครั้งเดียว จึงต้องทำการตรวจสอบปรับปรุงการทำงานของกระบวนงาน USER และ กระบวนงาน PASS

3.2 กระบวนงาน USER

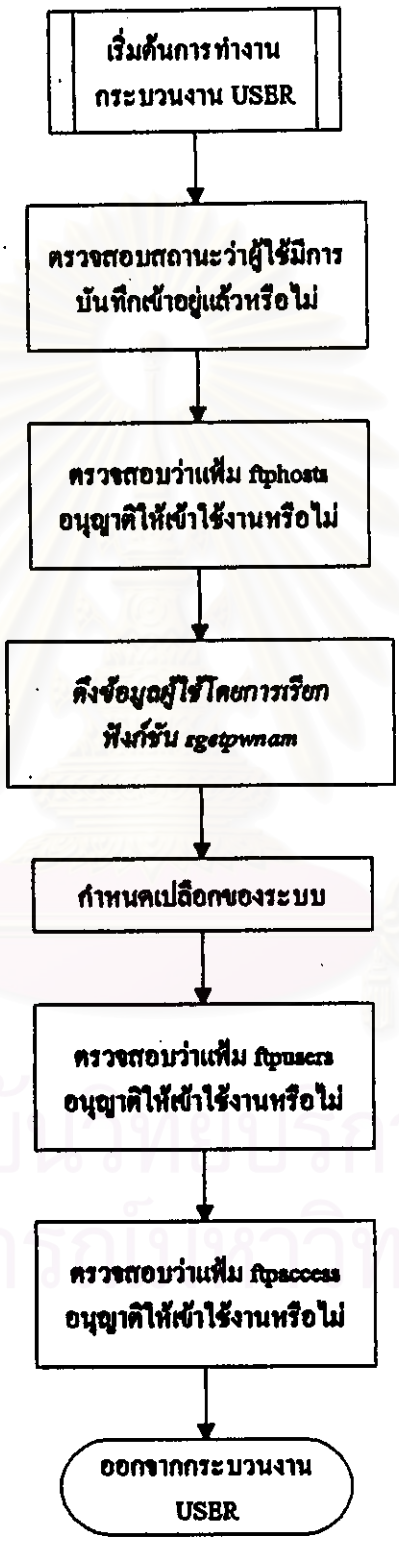
กระบวนงาน USER ทำหน้าที่ในการนำชื่อผู้ใช้มาประมวลผลเพื่อที่จะได้ทราบว่า ผู้ใช้มีสิทธิในการเข้ารับบริการถ่ายโอนแฟ้มหรือไม่ โดยในขั้นต้นจะมีผู้ใช้หลักๆ อยู่ 2 ประเภท

- ผู้ใช้ประเภทอาคันตุกะ (guest) ได้แก่ ผู้ใช้ชื่อ ftp และ ผู้ใช้ชื่อ anonymous
- ผู้ใช้ที่มีรายชื่ออยู่ในแฟ้ม /etc/passwd

กระบวนงาน USER มีรูปแบบการเรียกใช้ดังต่อไปนี้

```
void user ( char *name )
```

โดยที่ อากิวเมนต์ name คือชื่อลงบันทึกของผู้ใช้ ซึ่งกระบวนงาน USER จะต้องสามารถรองรับผู้ใช้ทั้งสองประเภท โดย ในรูปที่ 3.4 จะแสดงขั้นตอนการทำงานของกระบวนงาน USER จะเห็นได้ว่าการดึงข้อมูลของผู้ใช้ สามารถทำได้โดยการเรียกฟังก์ชัน sgetpwnam ดังนั้น หากต้องการให้โปรแกรม wu-ftpd สามารถใช้งานกับระบบยูนิกซ์หลายๆรุ่น จะต้องทำการปรับปรุง ฟังก์ชัน sgetpwnam เพิ่มเติม



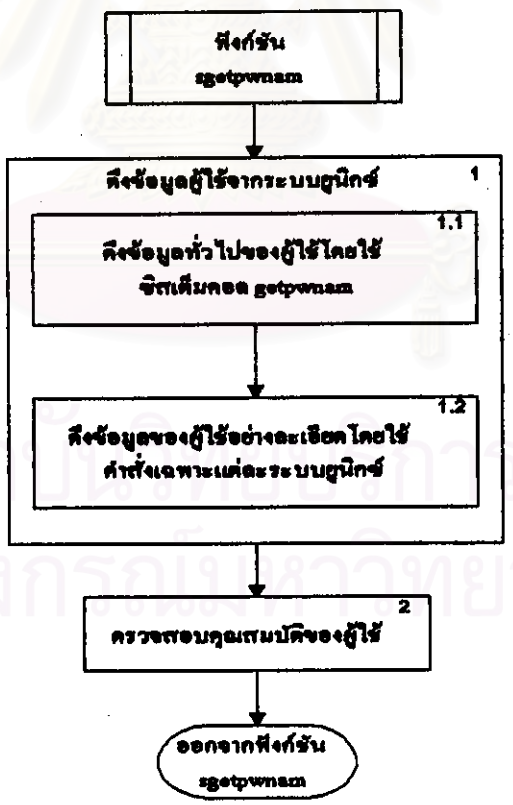
รูปที่ 3.4 แสดงการทำงานของกระบวนงาน USER

3.3 ฟังก์ชัน sgetpwnam

เป็นฟังก์ชันที่ต้องปรับปรุงเพื่อให้

- สามารถดึงข้อมูลของผู้ใช้
- สามารถใช้กับระบบรักษาความปลอดภัยหลายระบบ
- สามารถรองรับคุณสมบัติพิเศษของระบบยูนิกซ์หลายระบบ

โดยฟังก์ชัน sgetpwnam จะทำการเลียนแบบการทำงานของซิสเต็มคอล getpwnam ที่ใช้ได้เฉพาะกับระบบรักษาความปลอดภัยที่เก็บรหัสผ่านไว้ใน /etc/passwd เท่านั้น ดังนั้น ทั้งโครงสร้างของข้อมูลรับเข้า และ ส่งออก ของฟังก์ชัน sgetpwnam จะเหมือนกับ ซิสเต็มคอลทุกประการ เพื่อให้การเรียกใช้สามารถทำได้ง่าย และไม่ต้องมีการปรับเปลี่ยนวิธีการเรียกใช้ เมื่อต้องนำโปรแกรมไปใช้กับระบบยูนิกซ์แบบต่างๆ



รูปที่ 3.5 แสดงการทำงานของฟังก์ชัน sgetpwnam

จากรูปที่ 3.5 ข้อที่ 1.2 มีการดึงข้อมูลผู้ใช้โดยละเอียด ซึ่งเกิดขึ้นเนื่องจากแต่ละระบบยูนิกซ์มีระบบรักษาความปลอดภัยที่ต่างกันไป จึงไม่สามารถใช้คำสั่งดึงข้อมูลผู้ใช้แบบธรรมดา (getpwnam()) สำหรับทุกระบบรักษาความปลอดภัย ดังจะกล่าวถึงอีกครั้งในบทที่ 4 การพัฒนาโปรแกรม

ฟังก์ชัน sgetpwnam() มีรูปแบบของโครงสร้างข้อมูลและการเรียกใช้ดังต่อไปนี้

```
#include <pwd.h>
struct passwd *sgetpwnam(char *name)
```

โดยที่ name คือ ชื่อลงบันทึกเข้าใช้ของผู้ใช้ที่ต้องการค้นหาข้อมูล และเมื่อเรียกใช้ฟังก์ชันนี้ ค่าที่ได้กลับมา คือ ข้อมูลของผู้ใช้ที่มีโครงสร้างดังนี้

```
struct passwd {
        char    *pw_name;       /* login-name */
        char    *pw_passwd;     / * encrypted-password */
        int     pw-uid;         /* user-ID */
        int     pw_gid;         /* group-ID */
        char    *pw_comment;    /* not used */
        char    *pw_gecos;      /* miscellany */
        char    *pw_dir;        /* login-directory */
        char    *pw_shell;      /* shell */
}
```

ซึ่งจะเห็นได้ว่ามีโครงสร้างและการเรียกใช้งานเหมือนกับซิสเต็มคอล getpwnam() ทุกประการ รวมทั้งหากหาข้อมูลของผู้ใช้ไม่พบ จะคืนค่า NULL เช่นเดียวกัน

3.4 กระบวนงาน PASS

เป็นกระบวนงานที่ต้องได้รับการปรับปรุงเพื่อให้

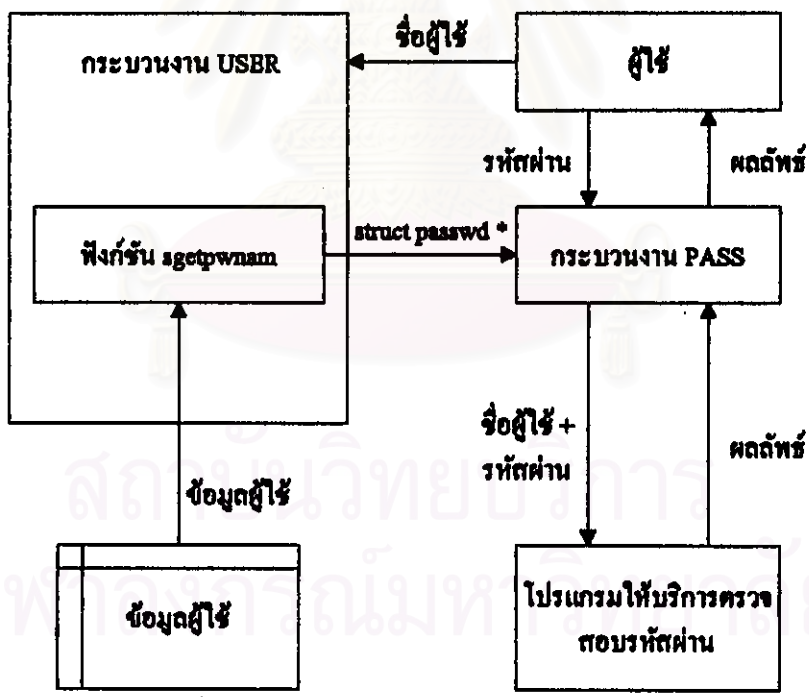
- ตรวจสอบรหัสผ่านว่าถูกต้องหรือไม่

- สามารถตรวจสอบรหัสผ่านของระบบยูนิกซ์หลายระบบ
- สามารถตรวจสอบรหัสผ่านแบบใช้ครั้งเดียว
- กำหนดค่าเริ่มต้นของสภาพแวดล้อมการทำงานให้แก่ผู้ใช้

โดยมีการเรียกใช้ดังต่อไปนี้

```
void pass ( char *password )
```

อาก้วเมนต์ password คือ รหัสผ่านที่ผู้ใช้พิมพ์ ในกรณีที่เป็นผู้ใช้ที่ใช้ระบบรหัสผ่านของระบบยูนิกซ์กระบวนงาน PASS จะนำข้อมูลของผู้ใช้ที่ได้จาก ฟังก์ชัน sgetpwnam () ของกระบวนงาน USER มาเปรียบเทียบ อาก้วเมนต์ password แต่หากเป็นผู้ใช้ที่ใช้ระบบรหัสผ่านแบบใช้ครั้งเดียว จะทำการนำอาก้วเมนต์ password และชื่อผู้ใช้ส่งไปตรวจสอบที่โปรแกรมให้บริการตรวจสอบรหัสผ่านแบบใช้ครั้งเดียว ดังรูปที่ 3.6



รูปที่ 3.6 แสดงการทำงานของกระบวนงาน PASS

โดยในรูปที่ 3.7 แสดงขั้นตอนการทำงานของ กระบวนงาน PASS ที่ได้รับการปรับปรุงแล้ว โดยจะเห็นได้ว่ามีการเรียกฟังก์ชัน request_otp_password เพื่อส่งรหัสผ่านไปยังโปรแกรมให้บริการตรวจสอบรหัสผ่านแบบใช้ครั้งเดียว

รูปที่ 3.7 แสดงขั้นตอนการทำงานของกระบวนงาน PASS

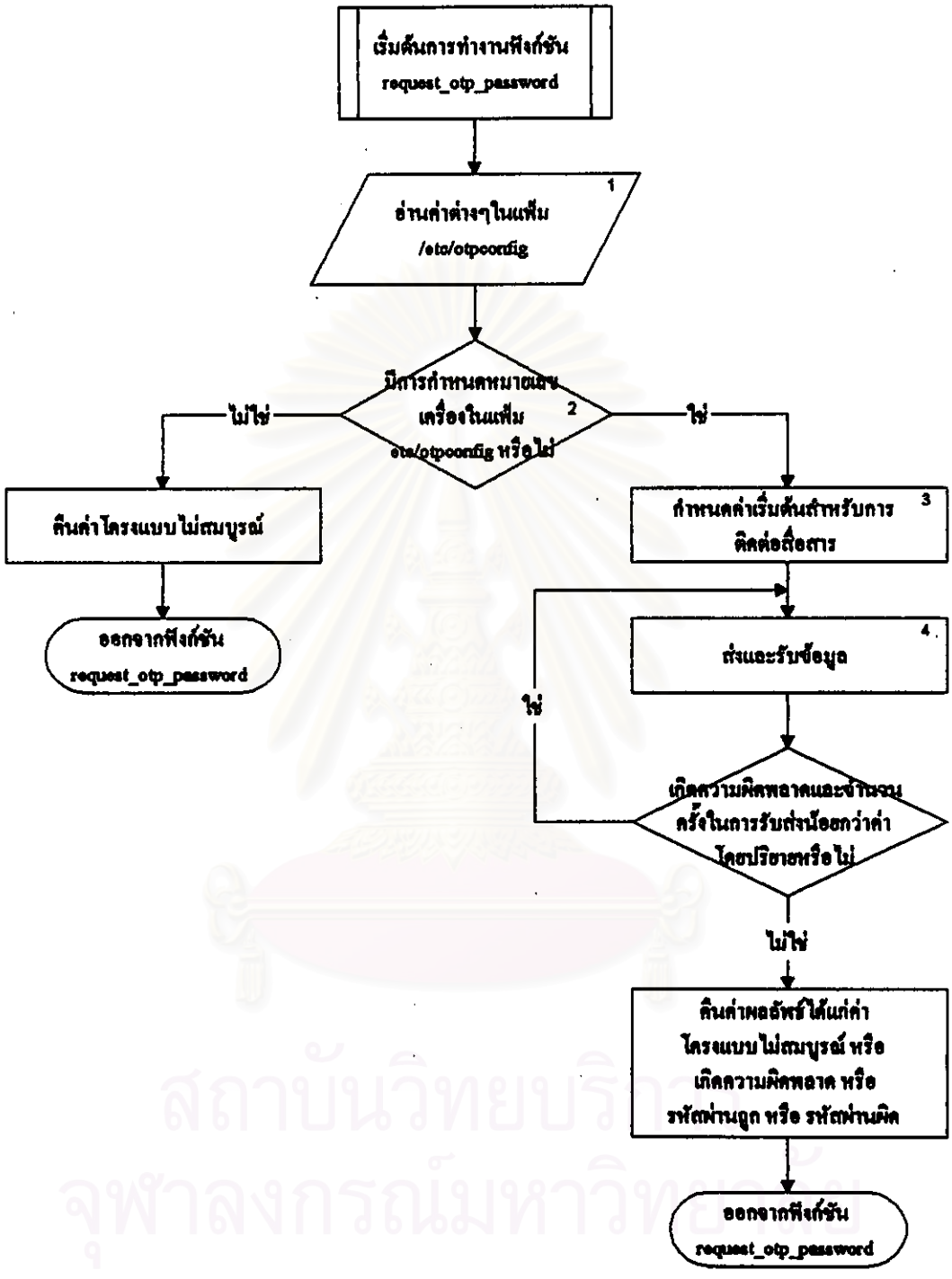
3.5 ฟังก์ชัน request_otp_password

ฟังก์ชัน request_otp_password() ถูกพัฒนาขึ้นเพื่อทำหน้าที่ส่งรหัสผ่านไปยังส่วนให้บริการตรวจสอบรหัสผ่านแบบใช้ครั้งเดียว โดยมีโครงสร้างและการเรียกใช้ดังต่อไปนี้

```
int     request_otp_password ( char *username, char *password )
```

โดยที่ username คือ ชื่อบันทึกเข้าของผู้ใช้ และ password คือ รหัสผ่านของผู้ใช้
ฟังก์ชัน request_otp_password() มีการคืนค่าดังต่อไปนี้

0	คือ รหัสผ่านผิด
1	คือ รหัสผ่านถูก
2	คือ ไม่สามารถติดต่อกับส่วนให้บริการตรวจสอบรหัสผ่านแบบใช้ครั้งเดียวได้
3	คือ โครงแบบ /etc/otpconfig ไม่สมบูรณ์
4	คือ มีการส่งข้อความจากโปรแกรมให้บริการตรวจสอบรหัสผ่าน



รูปที่ 3.8 แสดงขั้นตอนการทำงานของฟังก์ชัน request_otp_password

3.6 แฟ้ม /etc/otpconfig

เป็นแฟ้มที่ใช้ในการกำหนดค่าโครงแบบของฟังก์ชัน request_otp_password โดยที่ค่าต่างๆ ได้แก่

otphost คือ หมายเลขไอพีแอดเดรสของเครื่องให้บริการตรวจสอบรหัสผ่าน หรือ ใช้ชื่อเครื่องแทนก็ได้

port คือ ช่องทางสื่อสารของเครื่องให้บริการตรวจสอบรหัสผ่าน ค่าโดยปริยายคือ 6669

interval คือ เวลาในการส่งคำถามครั้งต่อไปหากไม่ได้รับการตอบกลับ ค่าโดยปริยายคือ 1

resend คือ จำนวนครั้งในการส่งคำถาม ค่าโดยปริยายคือ 1

หากเกิดความผิดพลาดในการกำหนดค่า ฟังก์ชัน request_otp_password จะใช้ค่าโดยปริยายในการติดต่อสื่อสาร แต่หากเกิดความผิดพลาดในเขตข้อมูล otphost ฟังก์ชัน request_otp_password จะทำการคืนค่า 3 (โครงแบบ /etc/otpconfig ไม่สมบูรณ์)

ในบทนี้ได้กล่าวถึงส่วนของโปรแกรมที่เกี่ยวข้องกับการพิสูจน์ตัวจริงทั้งหมดรวมถึงการออกแบบส่วนของโปรแกรม ได้แก่

- ฟังก์ชัน sgetpwnam()
- กระบวนงาน PASS
- ฟังก์ชัน request_otp_password()

ในบทต่อไปจะกล่าวถึงการพัฒนาส่วนของโปรแกรมต่างๆข้างต้น