บทที่ 3

ระบบ Time-Sharing (TSS)

ลักษณะการทำงานของ Hardware และ Software ในระบบ TSS

1. มีการติดต่อสื่อสารกันอย่างสม่ำเสมอกับอุปกรณ์ปลายทางในช่วงเวลาที่ดำเนินการ

2. เก็บโปรแกรมและข้อมูลในหน่วยช่วยเก็บความจำ (Secondary Storage) จนกระทั่งพร้อมที่จะใช้งาน

3. กำหนดลำดับ (Order) ของโปรแกรมและถ่ายทอดผลกลับไปสู่ปลายทางเฉพาะแห่ง

ในระบบ TSS ในการคำนวณเกี่ยวกับปัญหาใด ๆ หรือโปรแกรมใด ๆ ระบบการทำงานจะมีการแบ่งเวลา CPU เพื่อใช้ในงานหรือโปรแกรมต่าง ๆ วิธีการจัดระบบ Time-Sharing มี 2 ปัญหา คือ

1. การแบ่งส่วนความจำศูนย์กลาง (Main Memory)

2. การแบ่งเวลาศูนย์ประมวลผล (CPU)

การแบ่งส่วนความจำศูนย์กลางเป็นปัญหา คือ เนื่องจากพื้นที่ความจำศูนย์กลางมีขนาดจำกัด ดังนั้นโปรแกรมทุกโปรแกรมที่มีการแบ่งเวลา CPU ไม่อาจจะอยู่ในหน่วยความจำได้พร้อมกัน การแบ่งเวลา CPU เป็นเรื่องยาก เพราะว่า CPU จะต้องควบคุมและสั่งงานไปมาระหว่างโปรแกรมต่าง ๆ โดยการแบ่งเวลาให้แต่ละโปรแกรมเพื่อไม่ให้มีการรอคอยเป็นเวลานาน

การแบ่งส่วนความจำศูนย์กลางอาจแยกได้เป็น 2 แบบ

1. <u>Paging</u> เป็นวิธีการแบ่งโปรแกรมออกเป็นส่วน ๆ (Segment) ซึ่งแต่ละส่วนประกอบด้วยชุดของคำสั่ง (Sets of Instructions) ซึ่งเรียกว่า Pages ในแต่ละ Page ในงานทางด้านวิทยาศาสตร์ อาจถึง 1,000 คำคอมพิวเตอร์ (Computer Words)

เมื่อโปรแกรมถูก Run หลาย Page จากหลายโปรแกรมจะถูกป้อนเข้าไปในหน่วยความจำศูนย์กลางพร้อมกัน (Simultaneously) ด้วยวิธีการนี้ CPU จะทำงานไปมาระหว่าง Page ต่าง ๆ (ดังรูปที่ 14)

2. Memory-Swapping วิธีการนี้โปรแกรมทั้งโปรแกรมจะถูกย้าย (Move) เข้าไปเก็บหรือย้ายออกจากหน่วยความจำศูนย์กลาง โดยมีเพียงโปรแกรมเดียวเท่านั้นอยู่ในหน่วยความจำศูนย์กลางในระยะเวลาสั้นก่อนที่จะกระทำกับโปรแกรมอื่นในลักษณะเดียวกันต่อไป (ดูรูปที่ 15)

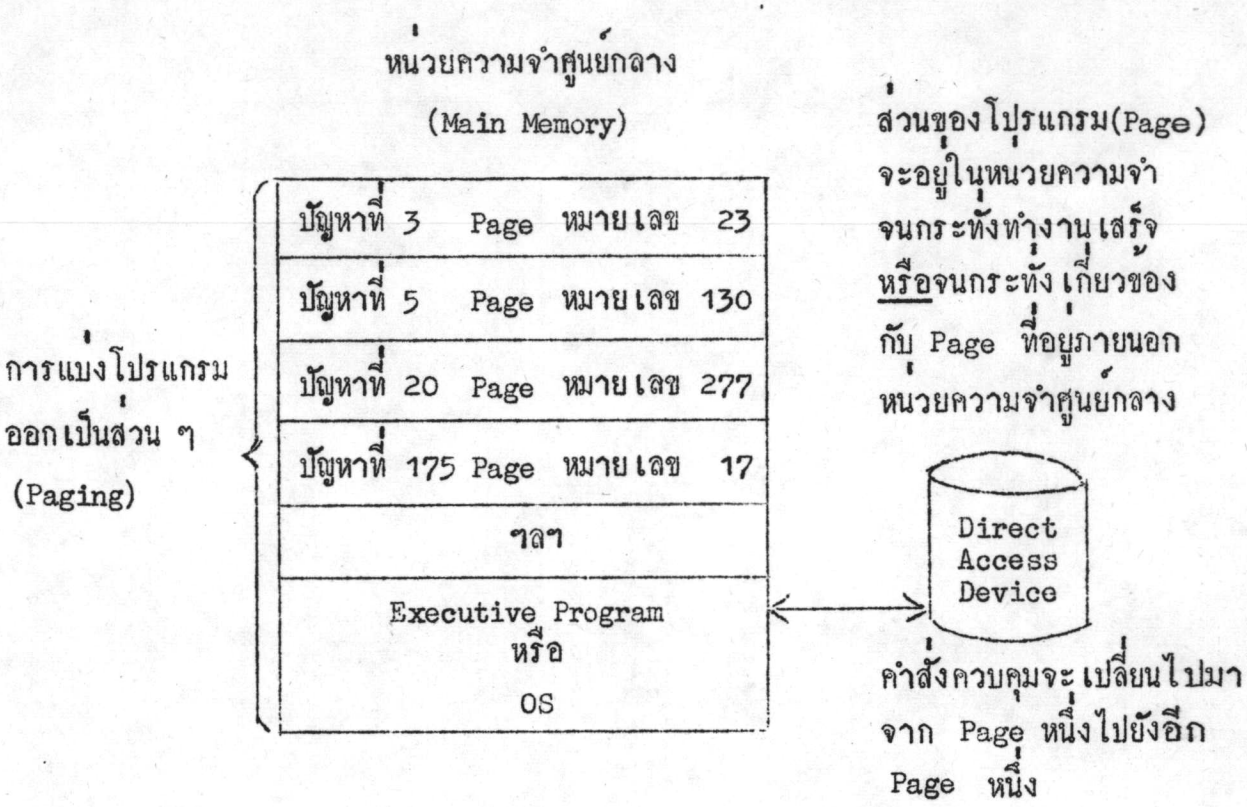
ทั้งสองวิธีที่กล่าวมานี้ต้องใช้โปรแกรมควบคุมการทำงาน และในการถ่ายทอดข้อมูลจากหน่วยช่วยเก็บความจำ (Secondary Storage) จะต้องมีอัตราเร็วสูง

การแบ่งเวลา CPU (Sharing the CPU)

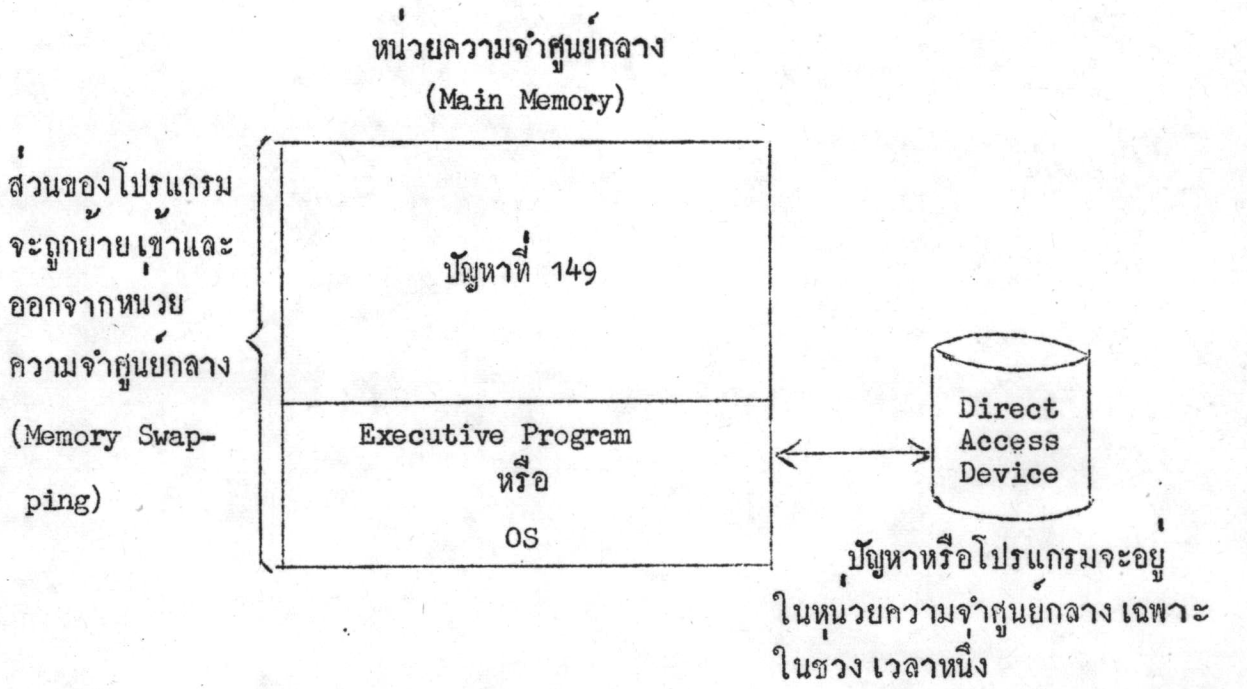
ถ้าหลาย Pages จากหลายโปรแกรมอยู่ในหน่วยความจำศูนย์กลางขณะเดียวกัน CPU จะแบ่งเวลาที่ใช้ในการประมวลผล (Processing Time) ระหว่างโปรแกรมเหล่านี้ การกำหนดว่า CPU จะทำงานอย่างไร มี 2 วิธี

วิธีที่ 1 การทำงานของ CPU จะเปลี่ยนไปยังโปรแกรมอื่นเมื่อสิ้นสุดช่วงเวลาอันสั้น คือแต่ละโปรแกรมจะได้ส่วนแบ่งเวลาแต่น้อย วิธีนี้เหมาะในงานด้านวิทยาศาสตร์ ซึ่งไม่ทราบเวลาการใช้ CPU

วิธีที่ 2 คือการทำงานของ CPU จะเปลี่ยนไปยังโปรแกรมอื่นทุกครั้งเมื่อโปรแกรมได้ทำงาน (Executed) เสร็จแล้วและรอคอยการเก็บผลใน Direct Access File หรือคอยการทำงานของหน่วย IO โปรแกรมหนึ่งอาจทำงานและรอคอยหลายครั้งกว่าจะทำงานเสร็จ ถ้าใช้วิธี Memory Swapping แต่ละโปรแกรมจะใช้เวลา CPU เพียงเล็กน้อยก่อนจะถูกย้ายออกไป แล้วส่วนของความจำส่วนนั้นจะถูกแทนที่ด้วยโปรแกรมอีกอันหนึ่งต่อไป ฉะนั้นโปรแกรมหนึ่งจึงมีการรอคอยให้ถึงรอบของตนเอง และจะมีการทำซ้ำแบบเดิม (Repeat The Cycle) จนกระทั่งทำงานเสร็จ



รูปที่ 14 แสดงถึงวิธีการ Paging



รูปที่ 15 แสดงถึงวิธีการ Memory Swapping

Time-Sharing Software

ระบบ TSS เหมือนกับระบบอื่น ๆ ที่กระทำหน้าที่ต่าง ๆ ตลอดจนหน้าที่ตามคำสั่งของโปรแกรม ซึ่งกำหนดให้ทำงาน Software Package โดยทั่วไปประกอบด้วย

1. Executive Programs หรือ Supervisory Programs เก็บไว้ในหน่วยความจำของระบบอย่างถาวร และให้บริการการดำเนินงานทุกอย่าง

2. User Programs ซึ่งอาจเขียนขึ้นเองโดยผู้ใช้หรือที่มีอยู่แล้ว โดยบริษัทผู้ผลิตสำหรับผู้ใช้ได้ใช้ในการแก้ปัญหา แต่ละโปรแกรมเหล่านี้มีหน้าที่โดยเฉพาะตามความจำเป็นในงานต่าง ๆ

Executive Programs หรือ Supervisory Programs มีหน้าที่ควบคุมระบบ TSS กำหนดหน้าที่ต่าง ๆ ในการดำเนินงานของระบบควบคุมความเร็วและความสามารถของคอมพิวเตอร์ ความสลับซับซ้อนขึ้นอยู่กับแบบและขนาดของระบบ TSS ซึ่งอาจมีความยาวของคำจาก 2 - 3 คำ ไปจนถึงเป็น 1000 คำ Executive Programs มีหน้าที่รับผิดชอบดังต่อไปนี้ คือ

1. จัดสรรพื้นที่ภายในหน่วยความจำและควบคุมสั่งงาน เพื่อการใช้ประโยชน์อย่างมีประสิทธิภาพเกี่ยวกับหน่วยความจำ (Main Memory) และหน่วยช่วยเก็บความจำ (Secondary Storage) Executive Programs จะจัดแบ่งส่วนของความจำสำหรับข้อมูลเพื่อการประมวลผล และควบคุมการ Relocation ของ User Programs ระหว่างการรอคอยและปฏิบัติงานอื่น ๆ มีตารางควบคุมการทำงานไว้เพื่อกำหนดตำแหน่งข้อมูล และป้องกันการเปลี่ยนแปลงแก้ไข (Transactions) ซึ่งอาจเกิดการเสียหายโดยบังเอิญหรือโดยจงใจ

Memory Protection เป็นสิ่งจำเป็นในระบบ TSS ทั่วไปที่มีผู้ใช้หลายคน ผู้ใช้บางคนอาจต้องการให้ข่าวสารหรือข้อมูลในแฟ้ม (Files) ของตนเป็นความลับ Executive Programs จะมีหน้าที่รับผิดชอบในการป้องกันพื้นที่ความจำที่กำลังใช้งานจากบุคคลอื่นที่ไม่เกี่ยวข้อง วิธีการง่าย ๆ คือ การร่วมมือใช้ระหัส (Code) ระหว่างผู้ใช้ที่กำหนดพื้นที่ความจำที่มีแฟ้มข้อมูล และอาจใช้ระหัสอื่น ๆ ด้วย โดยยอมให้ผู้ใช้อ่านได้แต่ไม่ยอมให้เปลี่ยนแปลงข้อมูลในแฟ้มข้อมูล (หรือตรงกันข้าม)

2. การจัดสรรเวลาคอมพิวเตอร์ (Computer Time Allocation) Executive Program จะควบคุมความต้องการผู้ใช้ ตลอดจนควบคุมอุปกรณ์ปลายทาง และจะกำหนดเวลา CPU ต่อผู้ใช้แต่ละแห่ง ช่วงเวลาของ CPU ที่ใช้จะขึ้นอยู่กับจำนวนผู้ใช้ทั้งหลายที่เกี่ยวข้อง และ Response Time ที่ผู้ใช้เหล่านั้นต้องการ หน้าที่ของ Executive Program คือ จะมีการ swap โปรแกรมเข้าและออกจาก Main Memory เพื่อบริการต่อผู้ใช้ที่จุดปลายทางทุกแห่งให้มีการเสียเวลาน้อยที่สุด

วิธีการหนึ่ง คือการกำหนด High-Priority Interrupt และ Low-Priority Interrupt สำหรับผู้ใช้ต่าง ๆ กัน

ดังนั้นกล่าวโดยย่อ Executive Program ไม่เพียงแต่มีหน้าที่จัดสรรเวลา CPU สำหรับงานต่าง ๆ แต่ยังกำหนด Priorities, Program Swapping และ Response to Interrupt จากเครื่องมือต่าง ๆ ในระบบและโปรแกรมต่าง ๆ ฯลฯ

3. ระบบ TSS จะทำงานทันทีที่มีข้อมูลมาจากปลายทาง (Polling Terminals) ในกรณีที่มีผู้ใช้พร้อมกันก็จะกำหนดลำดับก่อนหลัง (Priority) ของผู้ที่เรียก (Request) มาก่อนและหลังตามลำดับ และผู้ใช้อื่น ๆ ที่ต้องรอคอยให้ถึงรอบ (turn) Executive Program จะจัดการกับ Input ที่มาจากปลายทางโดยการ Scanning รายการของผู้ใช้ใน Memory ตาม Sequence

กล่าวโดยย่อ Executive โปรแกรมจะเลือก Input จากปลายทาง ถ้ามี Input Data มันจะสั่งให้โปรแกรมผู้ใช้ (User Program) ทำการประมวลผล หรือมิฉะนั้นมันจะเลือก Input ที่มาจากปลายทางอื่นตามลำดับ จนกระทั่งปลายทางทุกแห่งได้รับเลือกและได้รับบริการเสร็จแล้ว Executive Program จะควบคุมการทำงานของจุดปลายทางต่าง ๆ สัมพันธ์กับ User Program

4. การประมวลผลข้อมูลเข้าและข้อมูลออก ในระบบ TSS พร้อมด้วยวิธีการ Multiprogramming Executive Program จะควบคุมการประมวลผลของโปรแกรมหลายโปรแกรมเกี่ยวกับการจัดสรรเวลาของ CPU ปัจจุบันนี้ Executive Program มีขอบเขตหน้าที่กว้างขวางมาก

โปรแกรมของผู้ใช้ (User Programs)

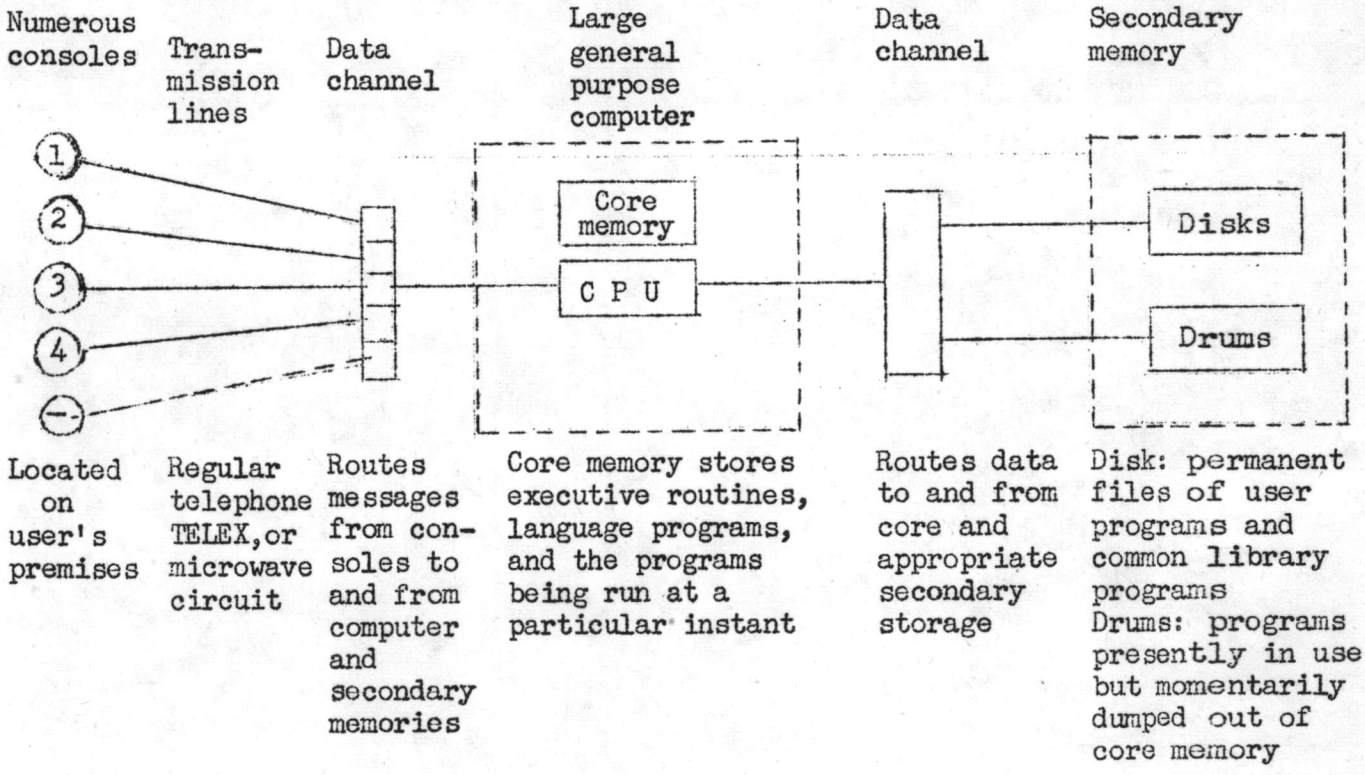
โปรแกรมผู้ใช้ คือ ชุดของคำสั่งที่สั่งให้ระบบคอมพิวเตอร์ทำงานในการแก้ปัญหาโดยเฉพาะ หรือทำตามคำสั่งที่กำหนดให้ เนื่องจากผู้ใช้ต้องการผลงานมากกว่าวิธีการที่ระบบทำงาน ดังนั้นภาษาที่ใช้กับระบบ TSS จึงได้มีการพัฒนาเพื่อจุดประสงค์อันนี้

ภาษาที่ใช้อย่างกว้างขวางใน Time-Sharing ที่เรียกว่า BASIC (ย่อมาจากคำว่า Beginners All-Purpose Symbolic Instruction Code) ซึ่งได้รับการพัฒนาที่ Darmouth College ระบบ TSS ที่ University of Hawaii (UHTSS) ซึ่งเป็นเครื่อง IBM OS/360 ในปี 1972 ภาษาที่ใช้กับระบบนี้คือ BASIC และ XPL (ซึ่งเป็นส่วนหนึ่งของภาษา PL/1) ผู้ใช้สามารถใช้ได้ง่ายและสะดวกเหมือนเครื่องคำนวณประจำโต๊ะ (Desk Calculator)

ภาษา XPL เป็นภาษาที่ใช้ได้กับงานทุกอย่าง และกับนักเขียนโปรแกรมทุกคน คือ รวมเอาส่วนดีที่สุดของภาษา FORTRAN และภาษา COBOL เข้าไว้ และมีลักษณะบางอย่างในตัวมันเองที่เป็นประโยชน์ ผู้ใช้ที่มีพื้นฐานการเขียนโปรแกรมจำกัดสามารถจะเรียนรู้ถึงวิธีการเขียนโปรแกรมได้โดยง่าย

ระบบ TSS ที่สถาบัน M.I.T.

ปี 1961 คณะกรรมการที่สถาบัน M.I.T. ได้พิจารณาเห็นว่า Turnaround Time ของระบบคอมพิวเตอร์ที่ใช้อยู่ขณะนั้นกินเวลานานมาก เป็นเหตุให้แผนกต่าง ๆ ของสถาบัน M.I.T. พยายามที่จะมีคอมพิวเตอร์ขนาดเล็กเป็นของตนเอง เมื่อพิจารณาถึงการที่มีคอมพิวเตอร์ขนาดเล็กโดยเปรียบเทียบถึงอัตราความสามารถกับค่าใช้จ่ายแล้ว ปรากฏว่าจะเพิ่มสูงขึ้นมากมาย จึงได้พิจารณาติดตั้งศูนย์คอมพิวเตอร์ระบบ TSS ที่มีขนาดใหญ่ขึ้นเพื่อสนองความต้องการของผู้ใช้ ปี 1963 จึงได้มีสัญญาร่วมกันระหว่างสถาบันกับ Advanced Research Agency of Department of Defense โครงการนี้มีชื่อว่า Project MAC ซึ่งในปี 1965 มีผู้ใช้ปลายทางถึง 300 คนใน 15 แผนก และห้องปฏิบัติการวิจัยหลายแห่ง



รูปที่ 16 ส่วนประกอบของระบบ TSS

Time-Sharing เป็นวิธีการทำงานของคอมพิวเตอร์ที่มีเครื่องมือ IO ปลายทาง อาทิเช่น Teletypewriters เชื่อมโยงกับสายโทรศัพท์ไปยังศูนย์กลางคอมพิวเตอร์ ระบบนี้ผู้ใช้เป็นจำนวนมากสามารถจะใช้ประโยชน์คอมพิวเตอร์พร้อม ๆ กันได้โดยเสรี ในเมื่อมีความจำเป็นต้องใช้เพื่อวัตถุประสงค์ต่าง ๆ ระบบ TSS ดังแผนภาพรูปที่ 16 มีลักษณะดังต่อไปนี้

1. ที่ศูนย์กลางคอมพิวเตอร์เป็นเครื่องขนาดใหญ่ใช้ได้กับงานทั่ว ๆ ไป (Large General Purpose Computer) อาจใช้ในงานที่เกี่ยวกับการเรียกใช้ข่าวสารหรือการคำนวณ ภาษาที่ใช้ได้กับระบบนี้คือ FORTRAN, COBOL และอื่น ๆ และมีความสามารถเก็บ Library Programs ได้เป็นจำนวนมาก

2. ระบบนี้จะให้บริการต่อผู้ใช้ทันที ผู้ใช้อาจจะใช้เมื่อไรก็ได้โดยการกดแป้น Console ป้อนโปรแกรมและข้อมูลเข้าไป ก็จะได้รับผลตอบสนองในเวลารวดเร็วเป็นเศษส่วนของวินาที

3. ผู้ใช้หลาย ๆ คนสามารถจะใช้อุปกรณ์ต่าง ๆ พร้อม ๆ กันได้ ขณะที่ผู้ใช้ปลายทางแห่งหนึ่งกำลังใช้อยู่จะไม่ถูกรบกวนจากการเรียก (Request) ใช้ของผู้ใช้ปลายทาง ณ จุดต่าง ๆ หรือทำให้รู้สึกว่าเวลาตอบสนอง (Response Time) ช้าลงไป

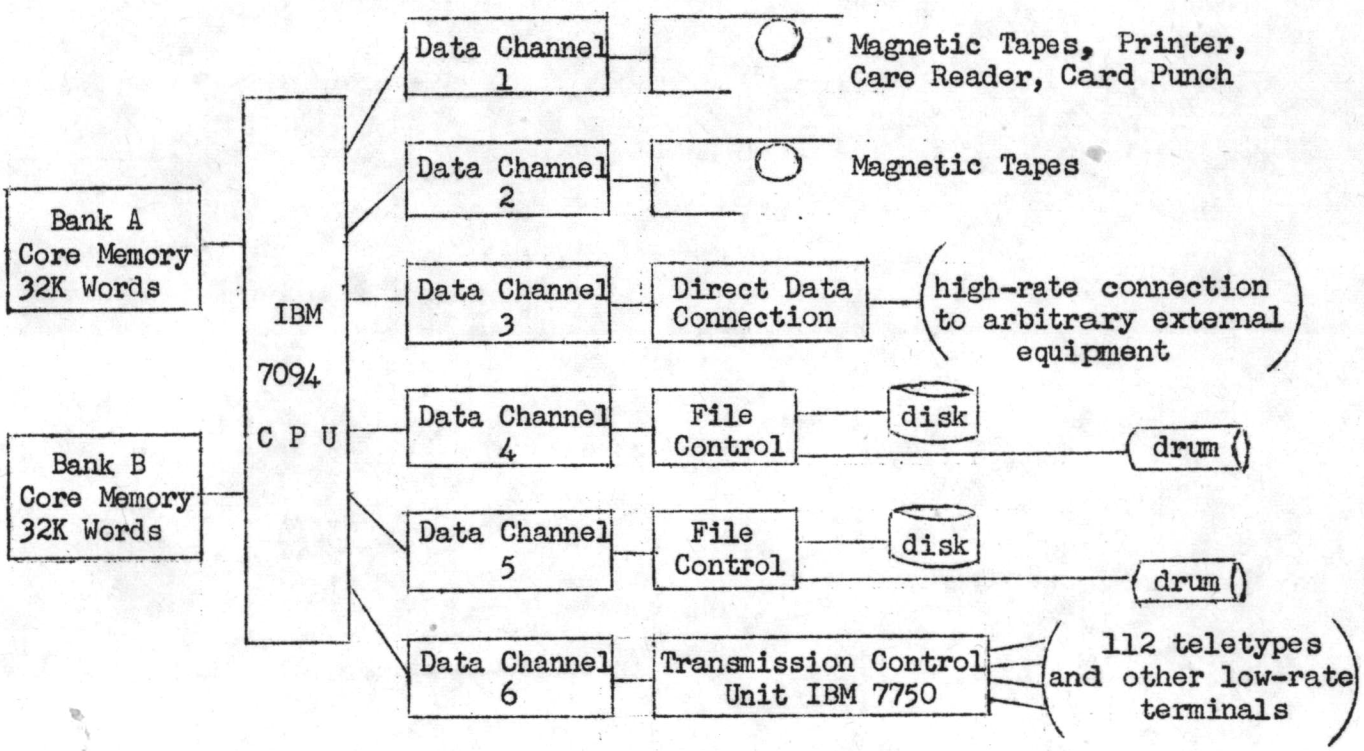
4. ทั้งผู้ใช้และระบบคอมพิวเตอร์มีความเป็นอิสสระ อาจจะใช้โปรแกรมต่าง ๆ บริการ และเครื่องมือ ซึ่งแยกจากกันได้โดยพร้อม ๆ กัน ผู้ใช้แห่งหนึ่งอาจใช้ภาษา FORTRAN ในขณะที่ผู้ใช้ปลายทางอีกแห่งหนึ่งใช้ภาษา COBOL

ระบบ TSS นี้สามารถให้ประโยชน์ต่อผู้ใช้เนื่องจากการทำงานของ Executive Programs ซึ่งมีอยู่ในระบบควบคุมการทำงานให้เกิดประสิทธิภาพ โดยการแบ่งเวลา CPU ในระยะเวลาสั้นให้แก่โปรแกรมผู้ใช้ปลายทางแต่ละแห่งโดยวิธี Multiprogramming

MAC System นี้ได้ใช้ในงานต่าง ๆ อย่างกว้างขวาง อาทิเช่น ในงานวิจัยทั้งภายในและภายนอกสถาบัน M.I.T. ตัวอย่างเช่น นักวิทยาศาสตร์ในแผนก Research Laboratory Electronics (RLE) ได้ใช้เป็นเครื่องมือช่วยในการศึกษาเรื่อง Plasma

Instability (Plasma หมายถึง ก๊าซซึ่งอยู่ในสภาพของอิออนที่เกิดขึ้น เนื่องจากลำแสงอิเลคตรอนที่มีความเข้มสูง ผ่านเข้าไปในศูนย์กลางของหลอดแก้วที่บรรจุก๊าซความดันต่ำ และอยู่ภายใต้สนามแม่เหล็กที่มีความเข้มสูง)

MAC System เป็นระบบ TSS ที่ใช้ IBM 7094 Central Processor ซึ่งได้มีการปรับปรุงให้มีระบบ Memory Protection และ Relocation ระบบนี้ประกอบด้วย Core Memory 2 หน่วย (ดังรูปที่ 17)



รูปที่ 17 ส่วนประกอบของ Project MAC System

แต่ละ Core Memory ประกอบด้วย 32,000 คำคอมพิวเตอร์ (Computer Words) หน่วยหนึ่งใช้สำหรับเก็บโปรแกรมผู้ใช้ต่าง ๆ (User Programs) ส่วนอีกหน่วยหนึ่งใช้สำหรับเก็บ Supervisory Program ซึ่งควบคุมการทำงานของระบบ TSS Secondary Storage ประกอบด้วย Drum 1 ตัว ซึ่งมีความจุ 185,000 คำ เพื่อเก็บโปรแกรมชั่วคราวในขณะที่โปรแกรมถูก swap ออกมาจาก Core Memory และ Disk 2 ตัว

มีความจุทั้งหมด 18 ล้านคำ ใช้ในการเก็บโปรแกรมหรือข้อมูลของผู้ใช้ต่าง ๆ ภาษาที่ใช้มีมากมาย อาทิเช่น FAP MAD FORTRAN COMIT LISP SLIP และภาษาที่ใช้ในงานโดยเฉพาะ ได้แก่ COGO และ STRESS

CPU ของระบบนี้ได้รับการปรับปรุงให้ทำงานแบบ Time-Sharing CPU จะทำงานเชื่อมโยงกับเครื่อง IO โดยผ่าน Channels ซึ่งทำหน้าที่ควบคุมข่าวสารหรือข้อมูลเข้าและออกจาก CPU ในระบบนี้มี 6 Channels 2 Channels ใช้สำหรับเครื่องมือ IO ได้แก่ Magnetics Tapes Printers Care Readers และ Card Punches Channels ที่ 3 ใช้สำหรับ Console ที่อยู่ปลายทางซึ่งต้องการการถ่ายทอดข้อมูลด้วยอัตราเร็วสูง ได้แก่ Oscilloscope Display Unit (CRT)

Channels ที่ 4 - 5 เชื่อมโยงกับ Disk และ Drum แยกกันต่างหากคนละ Channel แม้ว่าอัตราในการถ่ายทอดข้อมูลเข้า แต่มีความจุมาก Disk File สามารถเก็บได้ถึง 9,000,000 คำ และ Drum สามารถเก็บได้ถึง 185,000 คำ สำหรับ Disk Files มีความสามารถในการถ่ายทอดข้อมูลเข้าหรือออกจาก Core Memory ได้ 32,000 คำ ในเวลาประมาณ 2 วินาที Disk Files ใช้เก็บข้อมูล หรือโปรแกรมของผู้ใช้ Public Program และ Compiler ส่วน Drum ใช้เก็บโปรแกรมชั่วคราวในขณะเครื่องทำงาน

Channel ที่ 6 ใช้เชื่อมโยงกับเครื่องมือ IO ปลายทางกับระบบโดยผ่านหน่วยควบคุมการสื่อสาร ซึ่งทำหน้าที่เก็บโปรแกรม ควบคุมข้อมูลเข้าและออกจาก Teletype Consoles, Consoles เหล่านี้เชื่อมโยงกับระบบการถ่ายทอด TWX TELEX และ DATAPHONE

<u>สรุป</u> ที่กล่าวถึงส่วนประกอบของ Project MAC System นี้ต้องการแสดงให้เห็นถึงเครื่องมือต่าง ๆ ที่ใช้ในระบบ TSS โดยทั่วไปส่วนประกอบของระบบ TSS จะเหมือนกัน แต่อาจมีชื่อเรียกต่างกัน และอาจจะเพิ่มเติมหน่วยต่าง ๆ ที่เหมือนกัน โดยขึ้นอยู่กับความต้องการทางด้าน Core Memory หรือเครื่องมือ IO ตามความจำเป็นในการใช้งาน

ระบบ TSS ที่ University of Hawaii (UHTSS)

ปี 1968 เดือนกันยายน แผนกวิศวกรรมไฟฟ้าที่มหาวิทยาลัย Hawaii[17] ได้เริ่มโครงการที่มีชื่อว่า The Aloha System[18] เพื่อจะสำรวจหนทางเลือกการใช้สายในการสื่อสารในระบบคอมพิวเตอร์กระจายไปยังภาคพื้นต่าง ๆ ในการดำเนินงานที่มหาวิทยาลัย Hawaii ขณะนั้นใช้เครื่อง IBM 360/65 ซึ่งไม่ใช่ระบบ TSS โครงการ The Aloha System จึงได้ตัดสินใจสนับสนุนโครงการยอยด้วยวิธีการ Time-Sharing Facility เพื่อสนองความต้องการของผู้ใช้และมหาวิทยาลัยต่าง ๆ หลังจากได้สำรวจทางเลือกต่าง ๆ จึงได้ตกลงกันว่าไม่มีระบบไหนที่จะสนองความต้องการและข้อจำกัดต่าง ๆ ได้อย่างแท้จริง ดังนั้นระบบใหม่ที่มีชื่อว่า UHTSS (ยอมาจาก University of Hawaii Time-Sharing System) จึงได้รับการพัฒนา ซึ่งระบบนี้:-

1. ไม่รบกวนการดำเนินงานแบบ Batch
2. ขึ้นอยู่กับการควบคุมของ OS/360
3. ใช้ประโยชน์จาก OS /360
4. ต้องการจำนวน Main Memory พอสมควรซึ่งเป็นส่วนภายในระบบ และต้องการ เพิ่มขึ้นเล็กน้อย เมื่อผู้ใช้เพิ่มขึ้นแต่ละคน
5. ต้องการ Secondary Storage จำนวนจำกัด เพื่อควบคุม System Library และใช้เป็นตัวกลางสำหรับเก็บโปรแกรมหรือข้อมูลชั่วคราว เมื่อโปรแกรมหรือข้อมูลถูก Swap ออกมาจาก Memory โดยวิธีการ Memory Swapping

---

[17] Chadlie Carrol, Bass, Op. cit. ส่วนใหญ่ของเรื่องได้นำมาจากหนังสือเล่มนี้

[18] THE ALOHA SYSTEM is a research project at the University of Hawaii, supported by the Advanced Research Projects Agency of the Department of Defense and monitored by NASA Ames Research Center under Contract No.NAS2-6700, the U.S.Air Force Office of Aerospace Research (SRMA) under Air Force Contract No. F44620-69-C-0030 (a Project THEMIS award) and the National Science Foundation Under NSF Grant No. GJ-33220.

6. สามารถควบคุมได้ง่ายภายในมหาวิทยาลัย เพื่อที่นักศึกษาได้รับผลตอบแทนสูงตามที่กำหนดไว้ในโครงการ
7. กำหนดโครงสร้างพื้นฐานที่ Language Translator และอุปกรณ์ของผู้ใช้ต่าง ๆ ให้ใช้ร่วมกันได้ด้วยอย่างมีเหตุผล และ
8. สามารถออกแบบ, หรือสร้าง, เขียน, และทำให้เกิดผลภายใน 1 - 2 ปี

บริการที่ให้แก่ผู้ใช้ของ UHTSS มีดังนี้

1. การบำรุงรักษาแฟ้มข้อมูล ตลอดจนการพิมพ์ต่าง ๆ และคำสั่งให้แสดงผล
2. การเขียนโปรแกรมใช้ภาษา BASIC
3. การเขียนโปรแกรมใช้ภาษา XPL (ซึ่งเป็นส่วนหนึ่งของภาษา PL/1)
4. ลักษณะการดำเนินงานเสมือนเครื่องคำนวณประจำโต๊ะ
5. การสั่งงานหรือโปรแกรมเป็นแบบ Batch ที่ศูนย์กลางระบบ UHTSS
6. อุปกรณ์ที่ใช้ปลายทางทำงานในลักษณะ Batch Operation
7. ใช้ประโยชน์เพื่อ Copy, Concatenate และ Resequence Files

เหตุผลสำหรับระบบ TSS ส่วนมากต้องการที่จะให้บริการผู้ใช้พร้อมด้วยโปรแกรมที่ใช้ด้วยความรวดเร็วเพื่อสะดวกในการแก้ไขโปรแกรม สนับสนุนการทดลองพร้อมด้วยการคำนวณที่เกิดขึ้นปัจจุบันทันด่วน และใช้ประโยชน์คอมพิวเตอร์เสมือนเป็นเครื่องคำนวณประจำโต๊ะ สิ่งเหล่านี้เป็นหลักการในการพัฒนาระบบ UHTSS แต่ไม่ใช่วัตถุประสงค์ของระบบ UHTSS ที่จะให้บริการผู้ใช้ปลายทางที่จะใช้ความสามารถอย่างเต็มที่ของ OS /360 ซึ่งใช้ได้กับผู้ใช้งานแบบ Batch Operation เพื่อที่จะควบคุมได้ง่าย มีความเร็วและประสิทธิภาพสูงในการติดต่อระหว่าง UHTSS และผู้ใช้ (นั่นคือเกี่ยวกับระบบการควบคุม) และระหว่าง UHTSS และ OS/360 อยู่ในลักษณะที่สนับสนุนแนวความคิด Time-Sharing อันนี้ ดังได้กล่าวไว้ในหัวข้อเรื่องของระบบนี้

Logical Design

คำจำกัดความของหน้าที่ต่าง ๆ

หน้าที่ทางตรรกวิทยาสำหรับระบบ TSS อาจให้คำจำกัดความได้ดังนี้

(1) Management of System Activity

(2) Management of User and System Files

(3) Management of User Program Execution

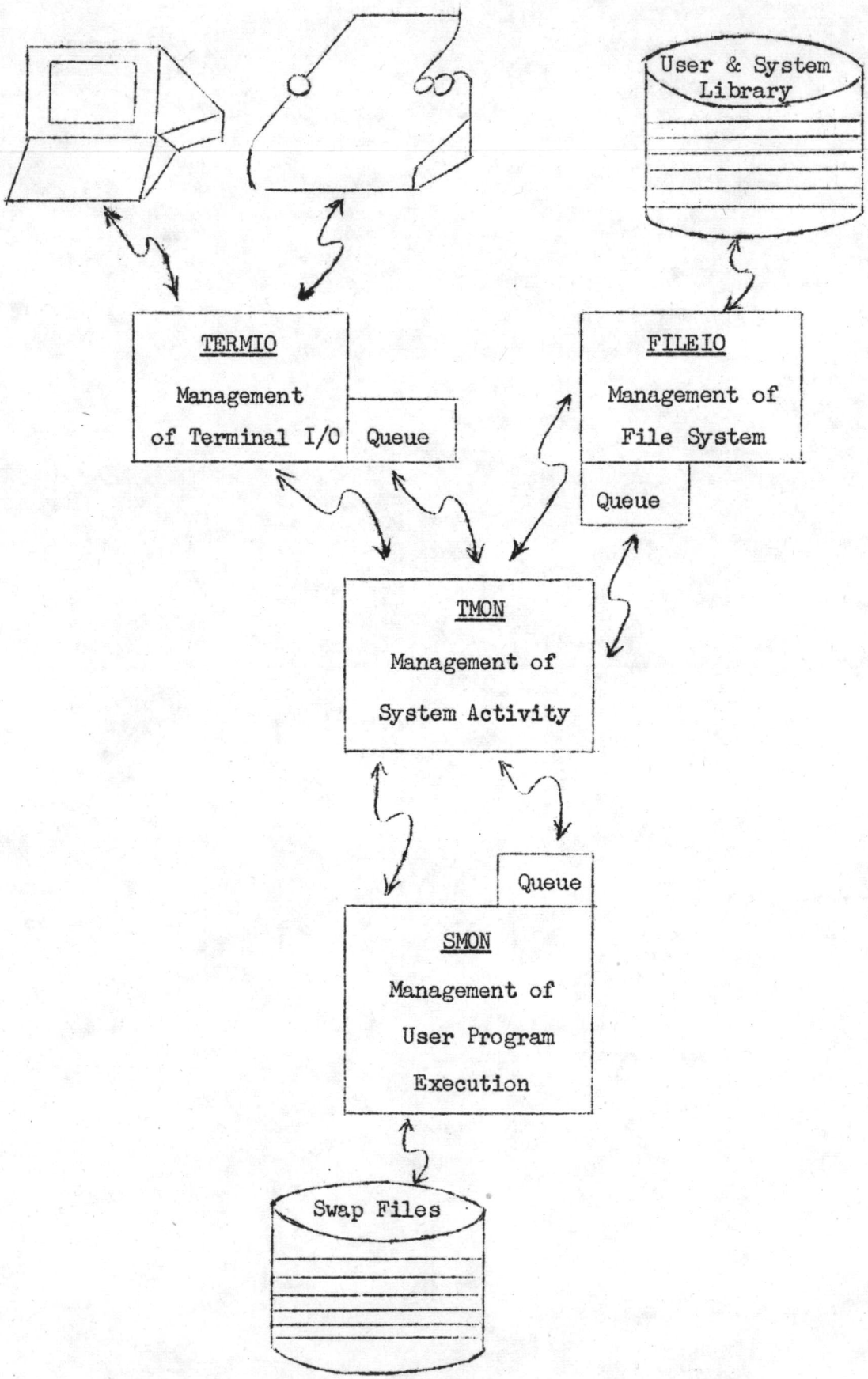
(4) Management of Terminal I/O

Modules ต่าง ๆ จะรับผิดชอบเกี่ยวกับหน้าที่ทางตรรกวิทยา (Logical Functions) เหล่านี้ ดังต่อไปนี้

1. การจัดการเกี่ยวกับกิจกรรมของระบบ (Management of System Activity)

Module ที่ทำหน้าที่นี้ชื่อว่า TMON เป็นตัวควบคุมการไหลของข่าวสารหรือข้อมูล (Communication Flows) ระหว่างส่วนต่าง ๆ ของระบบ UHTSS TMON จะถูกแจ้งให้ทราบว่า เมื่อไรผู้ใช้ปลายทางหรือความต้องการภายในระบบต้องการใช้บริการ แล้ว TMON จะกำหนดให้ว่า Module ไหนควรจะทำหน้าที่เกี่ยวกับงานที่สำคัญและกำหนด Entry เรียงตามลำดับงานสำหรับ Module ที่เหมาะสม แล้วต่อมา TMON ก็จะถูกบอกให้ทราบโดย Module ที่ทำงาน(ที่ถูกกำหนดให้) เสร็จเรียบร้อยแล้ว ซึ่งในเวลานั้น TMON ก็จะย้าย Entry อีกครั้งหนึ่งออกจากลำดับของงานและกำหนดว่าเมื่อไรลำดับการดำเนินงานอันต่อไปจะเริ่มต้น

รูปที่ 18 แสดงถึงการไหลของข่าวสารหรือข้อมูลระหว่าง Module ต่าง ๆ ของ UHTSS TMON กำหนดงานต่าง ๆ โดยเรียงลำดับก่อนหลัง Modules เหล่านี้เชื่อมโยงโดยตรงกับ TMON ในการทำงานร่วมกันนี้ TMON มี Priority ในการทำงานสูงที่สุด ด้วยเหตุนี้เมื่อไรที่ TMON ต้องการ CPU ก็จะสั่งงานไปยัง CPU หลังจากเสร็จงานที่ต้องการแล้วก็จะเลิกสั่ง TMON ไม่ได้ทำหน้าที่เกี่ยวกับหน่วย IO และหน้าที่อื่น



รูปที่ 18 Communication Flow Between Logical Modules

ที่ต้องการการรอคอยก่อนหรือหลัง การทำงานเกี่ยวกับกิจกรรมของระบบเป็นกระบวนการทำงานที่ตรงไปตรงมา คืองานแต่ละชิ้นจะเริ่มขึ้นและสำเร็จตามความต้องการก่อนที่จะกำหนดให้ทำงานต่อไปซึ่งไม่จำเป็นต้องเรียงลำดับก่อนหลัง Modules อื่น ๆ ทั้งหมดอาจจะถูกขัดจังหวะ (Interrupted) เมื่อไรก็ได้ และเริ่มต้นเมื่อไรก็ได้ อาทิเช่น หน่วย IO ซึ่งต้องการเป็นอิสระจาก CPU ดังนั้นงานต่าง ๆ อาจถูกกำหนดโดยไม่ขึ้นกับความสำเร็จของงานชิ้นก่อน ๆ และจำเป็นต้องมีการเรียงลำดับก่อนหลัง เพื่อจัดการเกี่ยวกับงานที่ได้รับมอบหมาย และอื่น ๆ

2. การจัดการเกี่ยวกับระบบแฟ้มข้อมูล (Management of File System)

การใช้ประโยชน์ OS/360 ระบบ UHTSS ควบคุมชุดข้อมูล OS/360 ซึ่งจัดสรรตำแหน่งและที่ว่างที่ต้องการสำหรับผู้ใช้และแฟ้มข้อมูลของระบบ Modules ซึ่งมีหน้าที่นี้ คือ FILEIO โดยใช้ OS/360 Disk เชื่อมโยงกับเครื่องมือ IO เพื่อว่าจะได้จัดสรรและการเรียกใช้ หรือบันทึกข้อมูลอย่างมีประสิทธิภาพ เมื่อมีการแจ้งให้ทราบจาก TMON FILEIO จะตรวจสอบลำดับก่อนหลัง (Queue) เพื่อจะกำหนดว่า ผู้ใช้ ณ จุดไหนต้องการบริการ รายละเอียดของการเรียก (Request) จะอยู่ใน Buffer

3. การจัดการเกี่ยวกับการทำงานของโปรแกรมผู้ใช้ (Management of User Program Execution)

วัตถุประสงค์เบื้องต้นของระบบ TSS ต้องการที่จะให้มีการทำงานของโปรแกรมและการเกี่ยวข้องกับผู้ใช้ปลายทางหลายแห่งในเวลาเดียวกัน Module ซึ่งควบคุมหน้าที่นี้มีชื่อว่า SMON เป็นหน่วยที่ทำหน้าที่ Swap โปรแกรมออกจาก Main Memory หรือเข้าไปสู่ Main Memory ในการแบ่งส่วนต่าง ๆ ของ Hardware ที่ต้องการใช้ SMON จะควบคุมลำดับก่อนหลังโปรแกรมของผู้ใช้ต่าง ๆ SMON จะบริการโปรแกรมผู้ใช้ปลายทางในลักษณะที่มาก่อนได้รับบริการก่อน การทำงานของโปรแกรมจะถูกขัดจังหวะหลังจากช่วงเวลาหนึ่ง ซึ่งเรียกว่า ส่วนแบ่งเวลา (Time Slice) แล้วโปรแกรมของผู้ใช้ซึ่งอยู่ในส่วนของ Main Memory จะถูกย้ายออกไปยัง Disk รอคอยเพื่อจะทำงาน

ต่อไป และแล้วโปรแกรมลำดับถัดไปก็จะถูกย้ายเข้าไปสู่ Main Memory เพื่อให้โปรแกรมทำงาน (วิธีการที่กล่าวนี้ เรียกว่า Memory Swapping) โปรแกรมที่รอคอยก็ถูกจัดลำดับให้อยู่ปลายแถว เพื่อคอยการทำงานเพิ่มเติมอีก

การทำงานของโปรแกรมอาจจะต้องรอคอยและถูกย้ายออกมาจาก Main Memory เมื่อผู้ใช้ปลายทางเรียกมา (Request) และอื่น ๆ

SMON ทำหน้าที่เกี่ยวกับหน่วย IO เมื่อมีการ Swapping และขึ้นกับ FILEIO ในการเรียกใช้หรือบันทึกข้อมูลส่วนใด ๆ ของผู้ใช้ หรือ System Library กับทั้งการ Loading ตัว Compiler, Utility Programs และอื่น ๆ

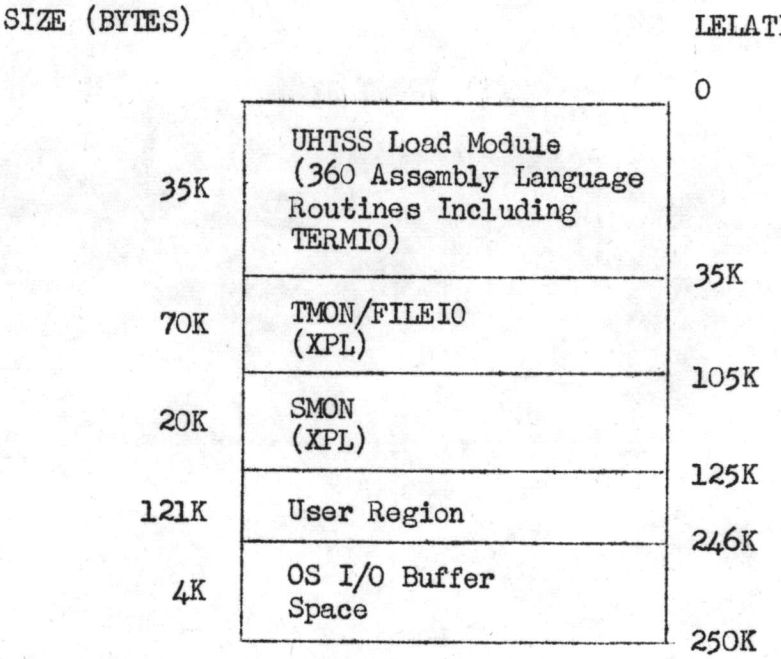
4. การจัดการเกี่ยวกับหน่วย IO ปลายทาง (Management of Terminal I/O

การสื่อสารกับผู้ใช้ปลายทางต่าง ๆ ควบคุมโดย Module ที่มีชื่อว่า TERMIO ปลายทางแต่ละแห่งจะมีแป้นสำหรับกด (Key) ส่งสัญญาณเรียกไปยัง OS/360 OS /360 ส่งสัญญาณตอบไปยังปลายทางและส่งสัญญาณ Interrupt ไปยัง Module ที่ชื่อว่า TERMIO เมื่อ TERMIO ประมวลผล Interrupt ก็จะย้ายไปยัง Buffer ที่มีชื่อว่า Surf และแจ้ง TMON ให้ทราบถึง Input ที่มาจากปลายทาง

สำหรับ Output ที่ไปสู่ปลายทาง Output Image จะถูกเก็บไว้ใน Surf Buffer แล้ว TMON จัดเข้าไปสู่ลำดับของ TERMIO Output นอกจากนี้ TERMIO ยังมีหน้าที่อื่น ๆ อีก

การจัดสรรพื้นที่ความจำศูนย์กลาง
(Memory Allocation)

รูปข้างล่างนี้แสดงถึงการจัดสรรพื้นที่ความจำศูนย์กลางภายในระบบ UHTSS มีหน่วยเป็น Kilo Bytes มีดังนี้ คือ



รูปที่ 19 แสดงถึง Memory Allocation

35K สำหรับ UHTSS Load Module (360 Assembly Language Routines Including TERMIO) 70K สำหรับ Module ที่มีชื่อว่า TMON และ FILEIO 20K สำหรับ Module ที่มีชื่อว่า SMON 121K สำหรับ User Region เพื่อใช้ในการทำงานของโปรแกรมของผู้ใช้ต่าง ๆ, 4K สำหรับ I/O Buffer Space

ระบบ UHTSS แบ่งการทำงานออกเป็น Module ต่าง ๆ ดังได้กล่าวมาแล้ว ซึ่งจัดว่าเป็นระบบที่ก้าวหน้า นั่นคือหน่วยอื่น ๆ สามารถจะต่อหรือรวมกับ Modules ที่มีอยู่เดิมได้

ภาษาที่ใช้กับระบบ UHTSS

ภาษาที่ใช้กับระบบนี้ ใช้ภาษา BASIC ซึ่ง BASIC Compiler ที่ใช้ตลอดทั้งระบบ UHTSS ได้เขียนขึ้นโดย Dr. W. W. Peterson เราอาจจะใช้ได้ง่าย ด้วยการเริ่มต้นพิมพ์ "EX BASIC" Compiler ก็จะถูกป้อนเข้าไป และ Input ก็จะถูกเรียก หลังจากนั้นก็สามารถป้อนคำสั่งในโปรแกรมเข้าไปตามกฎเกณฑ์ของภาษาที่ใช้

และอีกภาษาหนึ่งคือ ภาษา XPL XCOM ซึ่งเป็น XPL Compiler ใช้ได้ตลอดกับระบบนี้ โดยการเริ่มต้นพิมพ์คำสั่ง

EXECUTE XPL.

ภาษานี้เป็นส่วนหนึ่งของภาษา PL/1 ซึ่งรวมเอาส่วนดีของภาษา FORTRAN และ COBOL และรวมทั้งส่วนดีของตัวมันเอง เหตุผลที่ระบบนี้เลือกภาษา XPL แทนภาษา PL/1 คือ

1. XPL เรียนรู้ได้ง่ายกว่า
2. โปรแกรมที่ใช้ภาษา XPL สามารถ Compile ได้เร็วกว่า
3. โปรแกรมที่ใช้ภาษา XPL ป้อนเข้าไป (Load) ได้เร็วกว่า
4. โปรแกรมที่ใช้ภาษา XPL Run ได้เร็วกว่า

และ ฯลฯ

สรุปผลของระบบ UHTSS

ระบบ UHTSS ได้บรรลุวัตถุประสงค์ในการอำนวยความสะดวกแบบ Time-Sharing ทำให้เกิดประสิทธิภาพและประสิทธิผล พร้อมด้วยข้อกำหนดต่าง ๆ สำหรับ The Aloha System และความต้องการของส่วนรวมของมหาวิทยาลัย ลักษณะอันที่ 1 ของ The Aloha System ซึ่งเกี่ยวข้องกับเครื่องมือวิทยุต่าง ๆ ที่เชื่อมโยงกับ Teletype ได้รับผลสำเร็จโดยการใช้ UHTSS ยิ่งกว่านั้นเจ้าของโครงการส่วนบุคคล การดำเนินงานของกลุ่มได้ใช้ประโยชน์ UHTSS ด้วยเหตุนี้ ในทางปฏิบัติระบบ UHTSS อาจจะกล่าวได้อย่างชัดเจนว่าได้รับความสำเร็จ ในด้านการวิจัยงานที่ทันสมัย ควรจะพิจารณาได้เพียงว่า เพิ่งเริ่มต้น ส่วนการวัดผลอย่างละเอียดเกี่ยวกับ Simulation และ Theoretical Modulling จำเป็นต้องกระทำเพื่อว่าการประเมินผลการดำเนินงานและความสามารถของระบบจะได้ผลสมบูรณ์มากกว่านี้