รายงาน

เรื่อง IRON FIST

คณะผู้จัดทำ

กลุ่ม 17 Section 3

นายกัณชาติ	อภิสิทธิสานนท์	583021124-9
น.ส.อัจฉริยา	ชาเคน	583021156-6
น.ส.จันทรกานต์	แก้วมั่น	583020382-2
น.ส.ณัฐริกา	กุลดิลก	583021132-0
นาย สรวิชญ์	ไชยเดช	583021151-6
นาย อนุสิษฐ	โพธิ์เย็น	583021154-0

ชั้นปีที่ 2

Ses.1 กลุ่มที่ 17

อาจารย์ที่ปรึกษา

รศ.ดร.จักรชัย โสอินทร์

รายวิชา วิชา322222 NETWORK I

สาขาเทคโนโลยีสารสนเทศและการสื่อสาร คณะวิทยาศาสตร์ ภาควิชาวิทยาการคอมพิวเตอร์

มหาวิทยาลัยขอนแก่น

หลักการและเหตุผล

ในปัจจุบันเครือข่ายสื่อสารคอมพิวเตอร์และอินเทอร์เน็ตเข้ามามีบทบาทในชีวิตประจำวันมากขึ้น มีการใช้งานเว็บไซต์มากมาย การซื้อของออนไลน์ รวมทั้งการดาวน์โหลดโปรแกรมและการเชื่อมต่อกับเครือข่ายอินเตอร์เน็ต ซึ่งปัญหาที่พบ ได้แก่ บางที อินเตอร์เน็ตไม่สามารถเข้าถึงเราได้ตลอดเวลา และทันท่วงที ด้วยเหตุนี้ คณะผู้จัดทำ ได้เล็งเห็นถึงความสำคัญและความจำเป็นในการติดต่อกันระหว่าง IP โดยที่ไม่ต้องใช้อินเตอร์เน็ต เช่น การเล่นเกมกันระหว่าง 2 IP โดยใช้เครือข่ายในวงแลน เชื่อมต่อผ่านทางสายแลน ซึ่งถือเป็นการใช้ทรัพยากรที่มีได้อย่างมีประสิทธิภาพ จึงได้มีการเขียนเกม และเครือข่ายเพื่อเชื่อมต่อระหว่าง IP 2 เครื่อง เพื่อศึกษา และพัฒนา ให้ความรู้แก่ผู้ร่วมรับฟังต่อไป

วัตถุประสงค์

1) ศึกษาหาความรู้ในหัวข้อที่ศึกษา
2) รู้จักวิธีเชื่อมต่ออย่างถูกต้อง
3) การแลกเปลี่ยนข้อมูลความรู้ ประสบการณ์ปัญหาและอุปสรรคในการใช้งานเครือข่าย
4) เพื่อเผยแพร่ความรู้ และแนะนำผู้เข้าร่วมรับฟัง ในเรื่องการจัดการเครือข่าย
5) เพื่อให้ผู้เข้าร่วมรับฟังสามารถนำความรู้ที่ได้รับไปประยุกต์ใช้ในการทำงานได้อย่างมีประสิทธิภาพ
6) การสานความสัมพันธ์และสร้างสรรค์กิจกรรมสมาชิกภายในกลุ่ม

ทฤษฎีที่เกี่ยวข้อง

IP ADDRESS

IP Address ย่อมาจากคำเต็มว่า Internet Protocol Address คือ หมายเลขประจำเครื่องคอมพิวเตอร์แต่ละเครื่องในระบบเครือข่ายที่ใช้โปรโตคอลแบบ TCP/IP ถ้าเปรียบเทียบก็คือบ้านเลขที่ของเรานั่นเอง ในระบบเครือข่าย จำเป็นจะต้องมีหมายเลข IP กำหนดไว้ให้กับคอมพิวเตอร์ และอุปกรณ์อื่นๆ ที่ต้องการ IP ทั้งนี้เวลามีการโอนย้ายข้อมูล หรือสั่งงานใดๆ จะสามารถทราบตำแหน่งของเครื่องที่เราต้องการส่งข้อมูลไป จะได้ไม่ผิดพลาดเวลาส่งข้อมูล ซึ่งประกอบด้วยตัวเลข 4 ชุด มีเครื่องหมายจุดขั้นระหว่างชุด เช่น 192.168.100.1 หรือ 172.16.10.1 เป็นต้น โดยหมายเลข IP Address ของเครื่องคอมพิวเตอร์แต่ละเครื่องจะมีค่าไม่ซ้ำกัน สิ่งตัวเลข 4 ชุดนี้บอก คือ Network ID กับ Host ID

Introduction to Socket Programming

Socket Programming เป็นวิธีการสื่อสารระหว่างโปรเซส (IPC – Inter Process Communication) วิธีการหนึ่งซึ่งทำให้โปรเซสสามารถติดต่อ รับ/ส่ง ข้อมูลกันได้ ซึ่งลักษณะของ socket จะเป็นการสื่อสารแบบ full duplex โดยช่องทางการสื่อสารเดียวกัน สามารถใช้ได้ทั้งรับและส่ง และสามารถทำได้พร้อมกัน Socket Programming ไม่ได้มีใช้งานเฉพาะ ระหว่างโปรเซสที่อยู่ต่างเครื่องกัน หรือใช้สื่อสารผ่านเครือข่ายเท่านั้น แต่ได้ออกแบบมาให้ใช้สื่อสารระหว่างโปรเซสซึ่งมี parent ร่วมกันหรือใช้แทนการสื่อสารแบบ pipe ระหว่าง parent/child process หรือ child process ที่มี parent ร่วมกันก็ได้

Socketpair

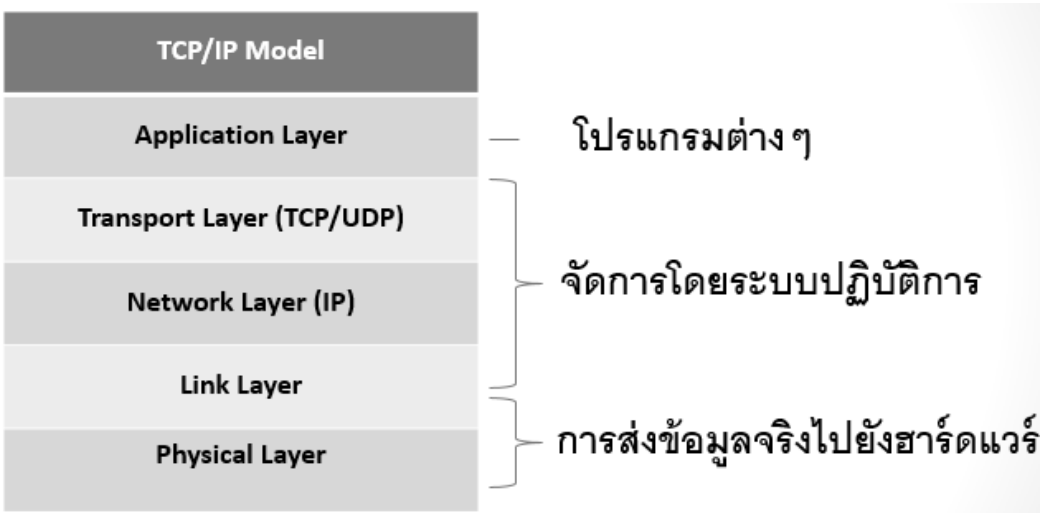
ตัวอย่าง การใช้การสื่อสารผ่าน socket ระหว่าง parent กับ child process โดยใช้ socketpair

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#include <sys/types.h>
#include <sys/socket.h>
#include <sys/wait.h>

#define BUFSZ   256
#define QSTR    "quit"

void    child(int s) {
        char    buf[BUFSZ];

        while (1) {
                ssize_t size;
                size=recv(s, buf, sizeof(buf), 0);
                if (size==-1) { perror("recv"); exit(-1); }
                buf[size]='\0';
                if (strncmp(buf, QSTR, strlen(QSTR))==0) {
                        printf("CHILD: got '%s' -- good bye\n", QSTR);
                        exit(0);
                } else {
                        system(buf);
                }
        }
}

int     main(void) {
        int     sv[2];
```

```
                        printf("> ");
                        fgets(buf, sizeof(buf), stdin);
                        buf[strlen(buf)-1] = '\0';    /* get rid of newline */

                        size=send(sv[1], buf, strlen(buf), 0);
                        if (size==-1) { perror("send"); exit(-1); }

                        if (strncmp(buf, QSTR, strlen(QSTR))==0) {
                                pid=wait(NULL);
                                if (pid==-1) { perror("wait"); exit(-1); }
                                printf("> child %d terminate\n", pid);
                                exit(0);
                        }

                }
        }
        return 0;
}
```

จากตัวอย่างโปรแกรมนี้ ใน main() จะสร้าง socket ที่ใช้สำหรับสื่อสารขึ้นมา โดยเรียกใช้ฟังก์ชัน socketpair หลังจากนั้นก็จะ fork เพื่อสร้าง child process และใช้ socket สำหรับส่งข้อมูลจาก parent process ไปยัง child process โดยตัว parent process จะรับข้อมูลที่ผู้ใช้พิมพ์เข้ามาแล้วส่งให้กับ child process โดยใช้ฟังก์ชัน send ในส่วนของ child process ก็จะรับข้อมูลจาก parent process โดยใช้ฟังก์ชัน recvแล้วส่งข้อมูลนั้นให้กับฟังก์ชัน system ซึ่ง เป็นการส่งข้อมูลที่ผู้ใช้พิมพ์ให้กับ shell ซึ่ง ผลของการ execute ของ shell จะแสดงผลออกทาง terminal ของ child process โดยตรง ซึ่ง ในที่นี้ก็จะเป็น terminal เดียวกับของ parent (child process ควรที่จะส่ง output ให้กับ parent กลับมาทาง socket !!)

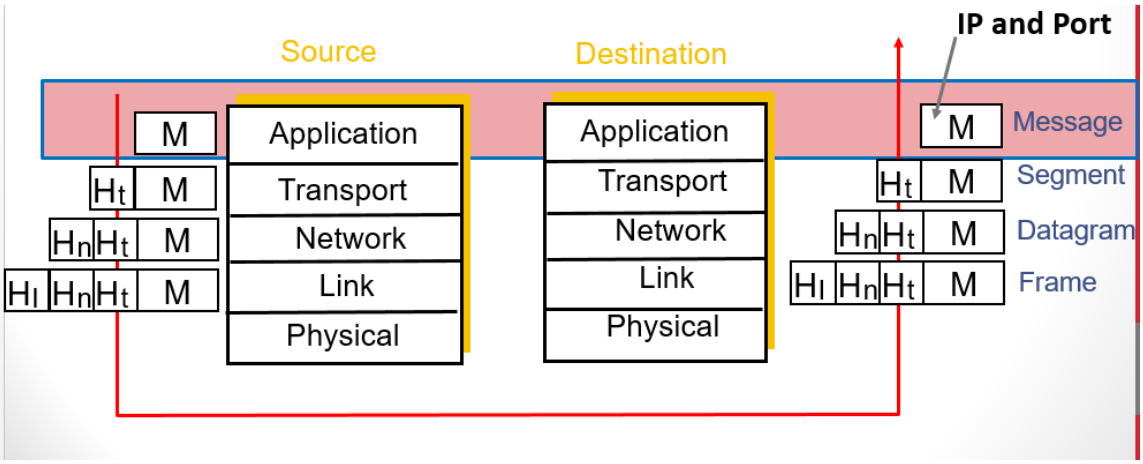
Network Protocol Stack



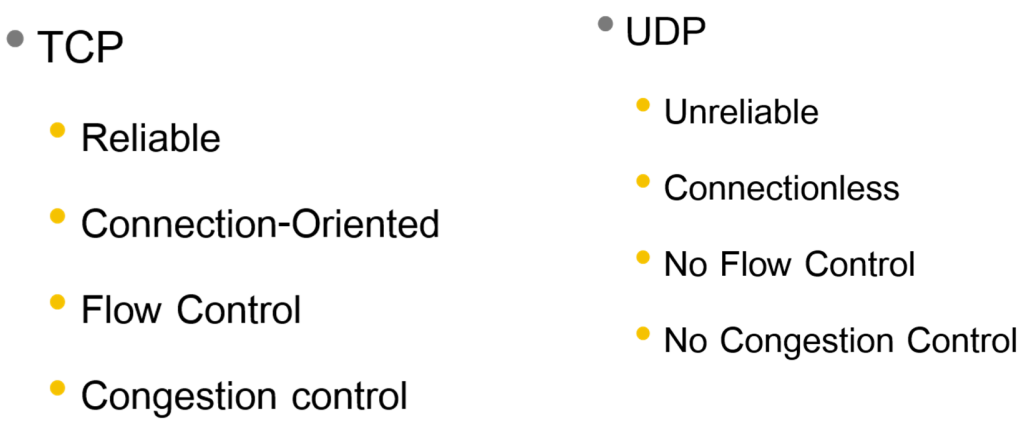
Network Protocol Stack (TCP/IP)

Protocol Layering and Data

ในแต่ละชั้นของโปรโตคอล จะมีการเพิ่ม Header เพื่อระบุรายละเอียดของข้อมูล (จัดการโดย OS)สิ่งที่สนใจในคลาสนี้คือ Application Layer



TCP and UDP



TCP

Reliable

ไว้วางใจได้ว่าข้อมูลที่ส่งไปจะถือผู้รับอย่างแน่นอน ซึ่ง TCP จะมีการตรวจสอบว่าข้อมูลที่ส่งไปนั้นถึงผู้รับจริง ๆ หรือไม่ ถ้าไม่ถึง TCP ก็จะทำการส่งข้อมูลนั้นไปให้ใหม่อีกครั้ง

Connection-oriented

มีการเชื่อมต่อช่องทางการรับส่งข้อมูลก่อนที่จะเริ่มส่ง เป็นการเตรียมทรัพยากรต่าง ๆ ระหว่างเครื่องผู้รับและผู้ส่ง เช่น socket (ดูเพิ่มเติมเรื่อง socket) หน่วยความจำ และตัวแปรที่ใช้เก็บข้อมูลต่าง ๆ เพื่อให้การรับส่งข้อมูลระหว่างต้นทางและปลายทางเป็นไปอย่างราบรื่นและมีประสิทธิภาพ

Flow control

มีการควบคุมปริมาณข้อมูลที่รับส่งระหว่างต้นทางและปลายทาง เพื่อป้องกันไม่ให้ฝั่งผู้ส่ง ส่งข้อมูลมากจนเกินกว่าที่ buffer ของฝั่งผู้รับจะรับได้

Congestion control

เป็นการควบคุมปริมาณการส่งข้อมูลเช่นกัน แต่เพื่อป้องกันไม่ให้ส่งข้อมูลเข้าไปในเครือข่าย ที่ ณ ขณะนั้นมีความหนาแน่นของข้อมูลสูงมาก ซึ่งมีความเสี่ยงที่ข้อมูลที่ส่งเข้าไปจะไปไม่ถึงผู้รับ ในทางตรงกันข้ามเมื่อเราหันมาดูคุณสมบัติของ UDP บ้าง จะเป็นดังนี้

UDP

Unreliable

ไม่รับประกันว่าข้อมูลจะถึงผู้รับหรือไม่

Connectionless

ไม่มีการสร้างช่องทางการรับส่งข้อมูลก่อนเริ่มส่ง

No flow control

ไม่มีการควบคุมปริมาณการรับส่งข้อมูลระหว่างต้นทางและปลายทาง

No congestion control

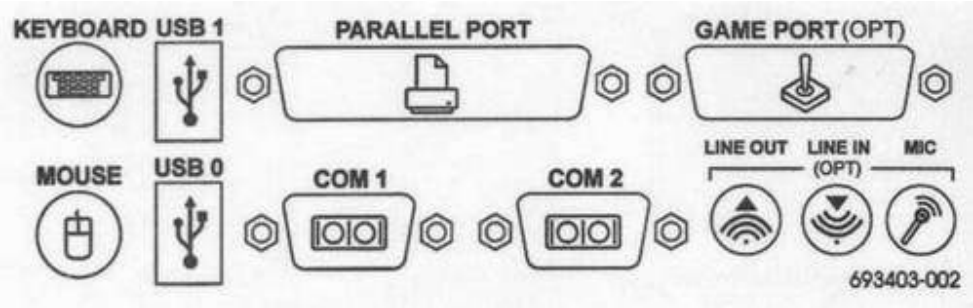
ไม่มีการควบคุมปริมาณการส่งข้อมูลระหว่างที่เครือข่ายมีความหนาแน่นสูง

IP, Port and Socket

IP address คือ เลขรหัสประจำคอมพิวเตอร์ที่ต่ออยู่บนเครือข่าย ซึ่งประกอบด้วยตัวเลข 4 ชุด และมีเครื่องหมายจุดขั้นระหว่างชุด ยกตัวอย่างเช่น 192.168.1.1 เป็นต้นหรือนิยมเรียกสั้นๆว่า IP ซึ่งตัวเลข IP แต่ละเครื่องจะไม่ซ้ำกัน ดังนั้น จึงได้มีการก่อตั้งองค์กรเพื่อ แจกจ่าย IP Address โดยเฉพาะ ชื่อองค์กรว่า InterNIC (International Network Information Center) อยู่ที่ประเทศสหรัฐอเมริกา การแจกจ่ายนั้นทาง InterNICจะแจกจ่ายเฉพาะ Network Address ให้แต่ละเครือข่าย ส่วนลูกข่ายของเครื่องทางเครือข่ายนั้นก็จะเป็น ผู้แจกจ่ายอีกทอดหนึ่ง ดังนั้นพอสรุปได้ว่า IP Address จะประกอบด้วยตัวเลข 2 ส่วน คือ

- Network Address
- Computer Address

พอร์ต (Port) เป็นช่องทางในการติดต่อสื่อสารระหว่างตัวคอมพิวเตอร์ กับอุปกรณ์ภายนอก ปกติพอร์ตจะอยู่ด้านหลังเครื่องคอมพิวเตอร์



Socket คือ กลุ่มของหมายเลข Port และ หมายเลข IP ซึ่งจะเป็นตัวบ่งชี้ที่เฉพาะเจาะจงสำหรับ Network process หนึ่งเดียวที่มีอยู่ในทั้งระบบ Internet คู่ของ Socket ที่ประกอบด้วย Socket หนึ่งตัว สำหรับต้นทาง และอีกตัว สำหรับปลายทาง สามารถใช้บรรยายถึงคุณลักษณะของ Connection oriented protocols

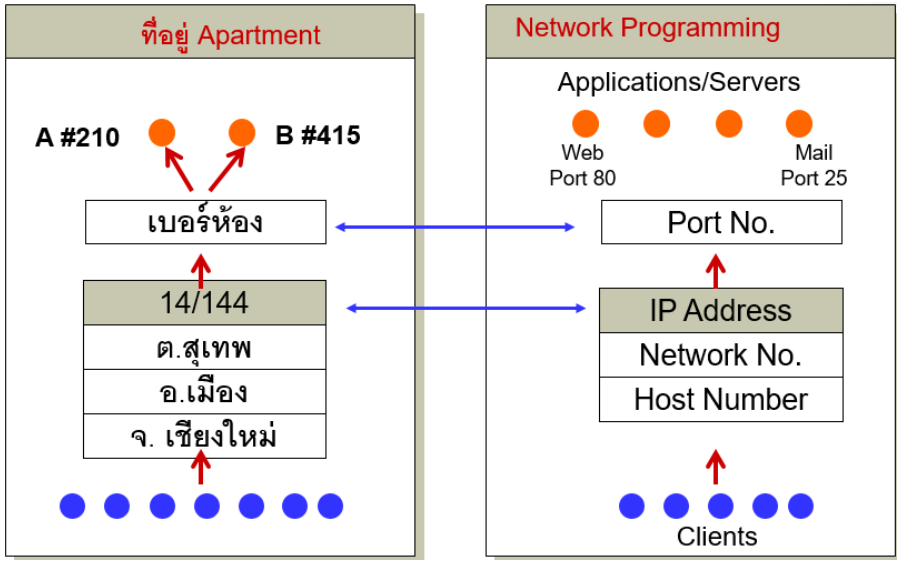
ข้อดี

Socket เป็นอะไรที่เป็นมาตรฐาน (แม้จะเก่าไปหน่อยก็ตาม) เพราะฉะนั้นการเปิด Socket *ไม่มี*ข้อจำกัดว่า Client, Server จะต้องเป็น platform เดียวกัน หรือภาษาเดียวกันจึงจะส่งได้

ข้อเสีย

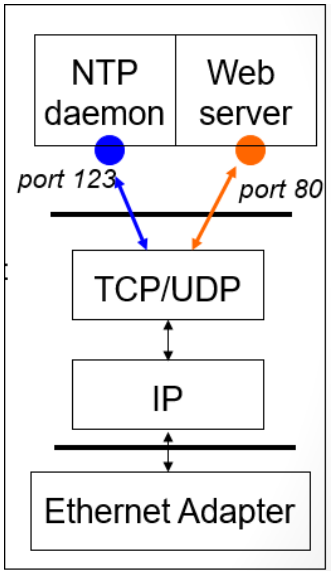
หากเราเปิด Socket เอง คงต้องเลือก port ดีๆ และไม่ควร hard code IP ของ Server หรือ port ลงไป แต่ควรจะเอาไปไว้ใน config file เพื่อให้สามารถปรับเปลี่ยนได้ง่ายในภายหลัง

IP Address and Port



IP Address and Port

Concept of Port Numbers



Concept of Port Numbers

เป็นการระบุที่อยู่ของ process ในแต่ละเครื่อง

Port numbers สามารถเลือกใช้ได้จาก

- Well-known (port 0-1023)
- Dynamic or private (port 1024-65535)

Servers ส่วนใหญ่จะมีการจอง Well-known port

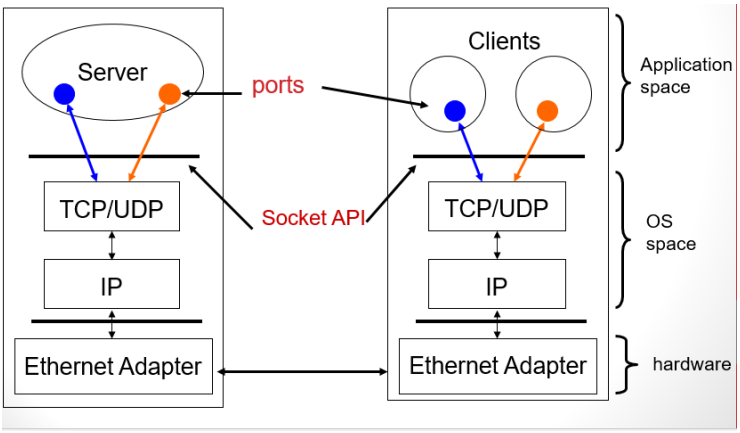
- Client ต่างๆจะได้รู้จัก
- HTTP = 80, FTP = 21, Telnet = 23, ...

Clientsส่วนใหญ่จะใช้ dynamic ports

- OS จะเป็นผู้กำหนดให้เอง

Client-Server Application

การติดต่อระหว่าง Client-Server จะผ่าน Socket API



Client-Server Application

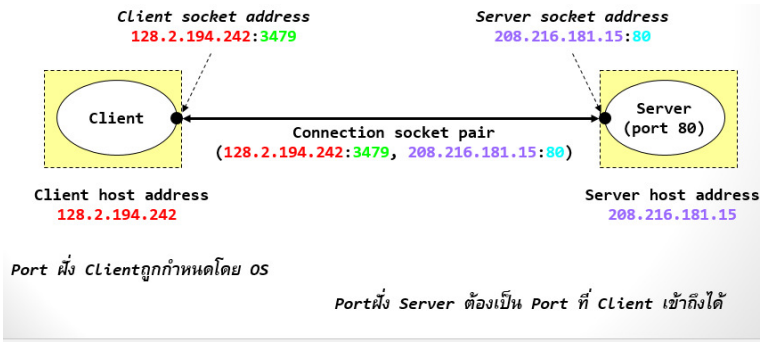
Socket Programming

Socket คือการเชื่อมต่อการสื่อสารระหว่างจุดสองจุด (เครื่องสองเครื่อง) แบบไปกลับได้ระหว่างโปรแกรมสองโปรแกรม (server กับ client) ภายในเครือข่ายเดียวกัน

- Server Socket Program เป็นโปรแกรมที่รันบนเครื่องที่มี socket ที่ผูกกับ Port number บนเครื่องและรอ request ที่จะเข้ามาจาก client
- Client Socket Program จะต้องรู้ว่า IP Address ของเครื่องที่ server socket program ทำงานอยู่ และ port ที่เครื่องนั้นรอฟัง request ด้วย

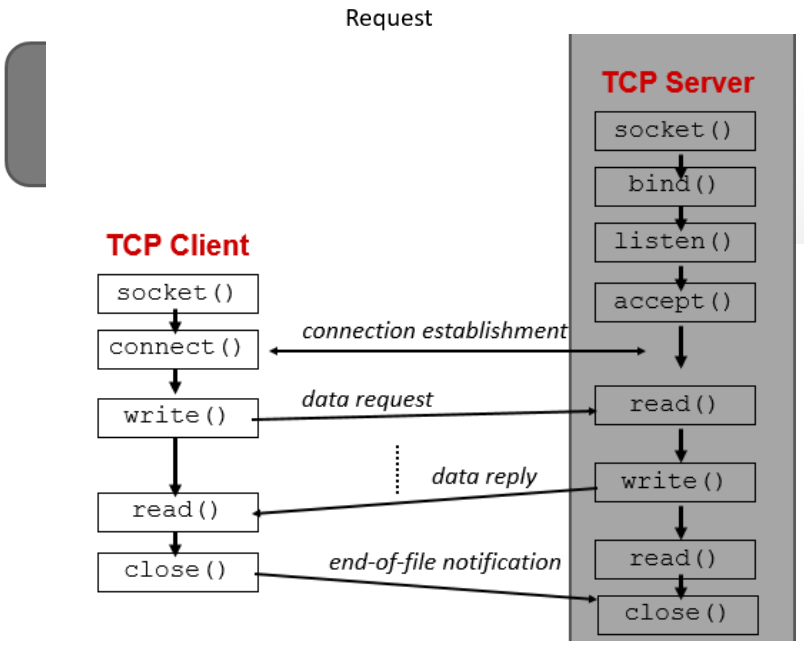
ส่วนประกอบของ Socket

- ที่อยู่ของเครื่อง (IP)
- ที่อยู่ของโปรแกรมในเครื่อง (Port)
- Address + Port=Socket



Client-Server Model

- เมื่อมีการสร้างการเชื่อมต่อระหว่าง server และ client ก็จะส่งข้อมูลผ่านทาง socket



TCP Client-Server Interaction

ชนิดของการเชื่อมต่อ

ในการเขียน Socket Programming ใน C# นั้นมี Protocol ในการเชื่อมต่อนั้นมี 2 ชนิด

- TCP/IP
- UDP/IP

TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหา โปรโตคอลก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้

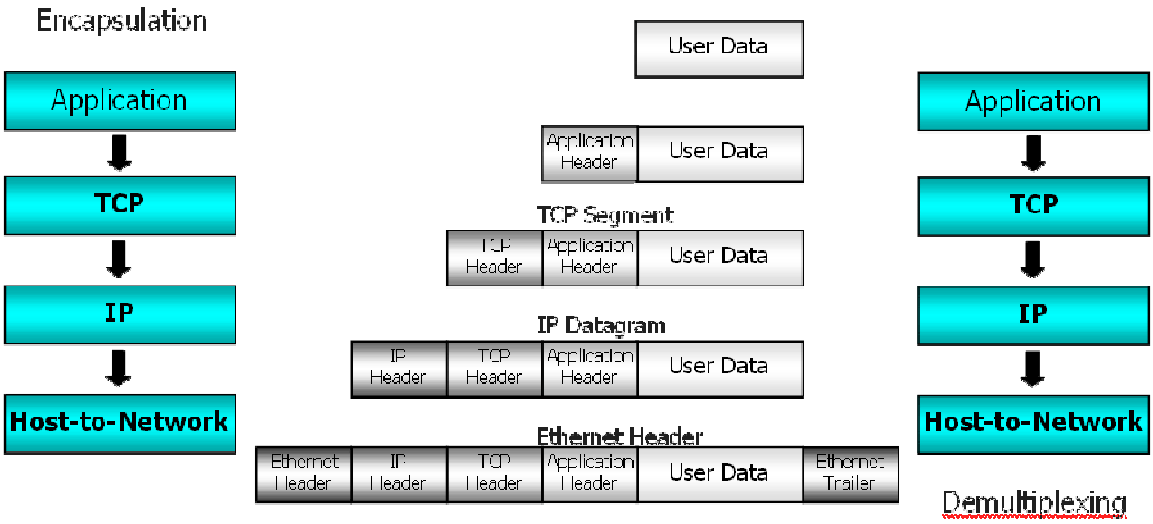
TCP/IP Protocol

TCP/IP มีจุดประสงค์ของการสื่อสารตามมาตรฐาน สามประการคือ

1. เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน
2. ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่นในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางทีใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ
3. มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ real-time และทั้งการสื่อสารแบบเสียง (Voice) และข้อมูล (data)

Encapsulation/DE multiplexing

การส่งข้อมูลผ่านในแต่ละเลเยอร์ แต่ละเลเยอร์จะทำการประกอบข้อมูลที่ได้รับมา กับข้อมูลส่วนควบคุมซึ่งถูกนำมาไว้ในส่วนหัวของข้อมูลเรียกว่า Header ภายใน Header จะบรรจุข้อมูลที่สำคัญของโปรโตคอลที่ทำการ Encapsulate เมื่อผู้รับได้รับข้อมูล ก็จะเกิดกระบวนการทำงานย้อนกลับคือโปรโตคอลเดียวกัน ทางฝั่งผู้รับก็จะได้รับข้อมูลส่วนที่เป็น Header ก่อนและนำไปประมวลและทราบว่าข้อมูลที่ตามมามีลักษณะอย่างไร ซึ่งกระบวนการย้อนกลับนี้เรียกว่า Demultiplexing



ขั้นตอนการ Encapsulation และ Demultiplexing

ชั้นสื่อสารนำส่งข้อมูล (Transport Layer)

แบ่งเป็นโพรโตคอล 2 ชนิดตามลักษณะ ลักษณะแรกเรียกว่า Transmission Control Protocol (TCP) เป็นแบบที่มีการกำหนดช่วงการสื่อสารตลอดระยะเวลาการสื่อสาร (connection-oriented) ซึ่งจะยอมให้มีการส่งข้อมูลเป็นแบบ Byte stream ที่ไว้ใจได้โดยไม่มีข้อผิดพลาด ข้อมูลที่มีปริมาณมากจะถูกแบ่งออกเป็นส่วนเล็กๆ เรียกว่า message ซึ่งจะถูกส่งไปยังผู้รับผ่านทางชั้นสื่อสารของอินเทอร์เน็ต ทางฝ่ายผู้รับจะนำ message มาเรียงต่อกันตามลำดับเป็นข้อมูลตัวเดิม TCP ยังมีความสามารถในการควบคุมการไหลของข้อมูลเพื่อป้องกันไม่ให้ผู้ส่ง ส่งข้อมูลเร็วเกินกว่าที่ผู้รับจะทำงานได้ทันอีกด้วย

โปรโตคอลการนำส่งข้อมูลแบบที่สองเรียกว่า UDP (User Datagram Protocol) เป็นการติดต่อแบบไม่ต่อเนื่อง (connectionless) มีการตรวจสอบความถูกต้องของข้อมูลแต่จะไม่มีการแจ้งกลับไปยังผู้ส่ง จึงถือได้ว่าไม่มีการตรวจสอบความถูกต้องของข้อมูล อย่างไรก็ตาม วิธีการนี้มีข้อดีในด้านความรวดเร็วในการส่งข้อมูล จึงนิยมใช้ในระบบผู้ให้และผู้ใช้บริการ (client/server system) ซึ่งมีการสื่อสารแบบ ถาม/ตอบ (request/reply) นอกจากนั้นยังใช้ในการส่งข้อมูลประเภทภาพเคลื่อนไหวหรือการส่งเสียง (voice) ทางอินเทอร์เน็ต

UDP : (User Datagram Protocol)

เป็นโปรโตคอลที่อยู่ใน Transport Layer เมื่อเทียบกับโมเดล OSI โดยการส่งข้อมูลของ UDP นั้นจะเป็นการส่งครั้งละ 1 ชุดข้อมูล เรียกว่า UDP datagram ซึ่งจะไม่มีความสัมพันธ์กันระหว่างดาต้า

แกรมและจะไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูล

กลไกการตรวจสอบโดย checksum ของ UDP นั้นเพื่อเป็นการป้องกันข้อมูลที่อาจจะถูกแก้ไขหรือมีความผิดพลาดระหว่างการส่ง และหากเกิดเหตุการณ์ดังกล่าว ปลายทางจะได้รู้ว่ามีข้อผิดพลาดเกิดขึ้น แต่มันจะเป็นการตรวจสอบเพียงฝ่ายเดียวเท่านั้น โดยในข้อกำหนดของ UDP หากพบว่า Checksum Error ก็ให้ผู้รับปลายทางทำการทิ้งข้อมูลนั้น แต่จะไม่มีการแจ้งกลับไปยังผู้ส่งแต่อย่างใด การรับส่งข้อมูลแต่ละครั้งหากเกิดข้อผิดพลาดในระดับ IP เช่น ส่งไม่ถึง, หมดเวลา ผู้ส่งจะได้รับ Error Message จากระดับ IP เป็น ICMP Error Message แต่เมื่อข้อมูลส่งถึงปลายทางถูกต้อง แต่เกิดข้อผิดพลาดในส่วนของ UDP เอง จะไม่มีการยืนยัน หรือแจ้งให้ผู้ส่งทราบแต่อย่างใด

งานที่เกี่ยวข้อง

MakrukThai 2004

MakrukThai 2004 เป็นเกมส์หมากรุกไทย ที่มีรูปแบบและหลักการเล่นที่เหมือนเล่นบนกระดานจริงทุกประการ ผู้เล่นสามารถวางหมากและเดินหมากได้อย่างอิสระ พร้อมทั้งมีฟังก์ชั่นการปรับตั้งค่าในการเล่นที่ละเอียดครบถ้วน สามารถดูข้อมูลการเดินของหมากในแต่ละตัว

นอกจากนี้โปรแกรมยังสามารถเล่นผ่านอินเทอร์เน็ต (Internet) หรือเล่นกันภายในเครือข่าย LAN (Local Area Network) รวมถึงระบบ TCP/IP Network ได้อีกด้วย

งานของเราคืออะไร

IRON FIST คือเกมที่มีการเชื่อมต่อกันภายในเครือข่าย LAN และระบบ TCP/IP การที่คอมพิวเตอร์จะสามารถเล่นเกมร่วมกันนั้น ต้องมีการเชื่อมต่อกันภายในเครือข่าย LAN นั้นซึ่งจะมีความสะดวกมากขึ้น และยังสามารถเชื่อมต่อผ่านระบบ TCP/IP ได้อีกด้วย โดยเกมของเราใช้ TcpListener/Client ในการติดต่อระหว่างเครื่องสองเครื่องโดนมีเครื่องหนึ่งเป็น Server และ อีกหนึ่งเครื่องเป็น Client

ความสามารถของIRON FIST

- เชื่อมต่อเครือข่ายกับอีกเครื่องหนึ่งโดยไม่ต้องมีหนึ่งโดยไม่ต้องใช้ internet
- มีขั้นตอนไม่มากในการเข้าใช้
- เป็นเกมเล่นง่ายๆสามารถเล่นได้ทุกวัย

ขั้นตอนการใช้งาน IRON FIST

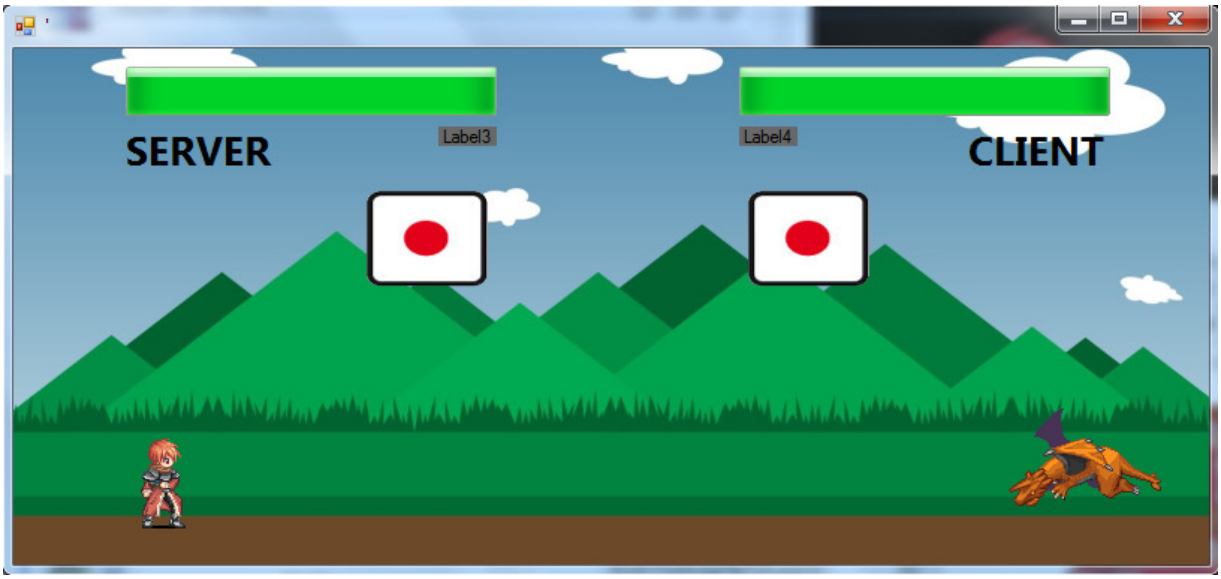
1.หน้าแรก เมื่อเข้าสู้โปรแกรมประกอบด้วยปุ่ม เริ่มเกม วิธีการเล่นเกม และ ปิดเกม

2.เมื่อกดปุ่มเริ่มเกม จะขึ้นหน้าให้เลือก ระหว่าง Server และ Client

3.ฝั่งทางซ้าย เป็นหน้าจอ สำหรับ เครื่อง Server และฝั่งทางขวาเป็นหน้าจอสำหรับ เครื่อง Client เมื่อใส่ IP ในเครื่อง Server แล้วคลิ้ก ปุ่ม OK หาก Client สามารถเชื่อมต่อได้ จะขึ้นสถานะเป็น Connected

ส่วน เครื่อง Client ให้ใส่ IP ของเครื่อง Server แล้วคลิ้กปุ่ม ok หากสามารถเชื่อมต่อได้จะขึ้นสถานะเป็น Connected

4. เมื่อกด ปุ่ม Start Game จะแสดงหน้าจอ ดังภาพ



วิธีการเล่มเกม IRON FIST

1. มีผู้เล่น 2 ฝ่าย คือ Server และ Client
2. ฝ่าย Server จะเป็นฝ่ายที่ได้เริ่มกดทอยลูกเต๋าก่อน ตามด้วยฝ่าย Client
3. หากฝ่ายใด ได้แต้มจากการทอยลูกเต๋ามากกว่า จะได้เป็นฝ่ายเคลื่อนที่เข้าไปโจมตี ฝ่ายตรงข้าม
4. เมื่อฝ่ายใดฝ่ายหนึ่ง สามารถทำให้HP ของฝ่ายตรงข้าม เท่ากับ 0 จะได้รับแต้ม 1 แต้ม
5. หากฝ่ายใด เก็บแต้มได้ครบ 2 แต้มก่อน จะเป็นฝ่ายชนะ