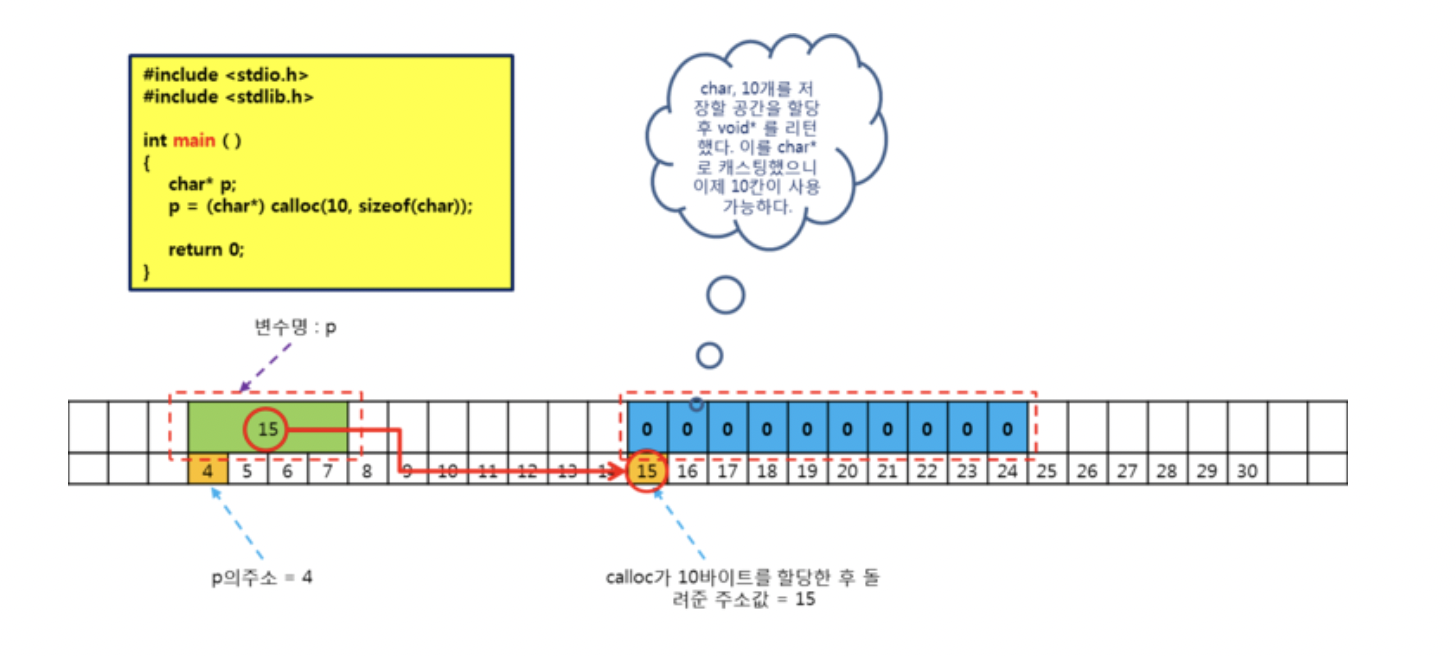
#1 calloc, realloc 조사하고 실습

#calloc

:동적할당메모리로 n개의 요소 배열에 대한 기억장치 공간을 예약.

:각 요소의 모든 비트에 초기값 0.

:void\* calloc (size\_t, size\_t size);



#include <stdio.h>

#include <stdlib.h>

**int** main(){

**char**\* p;

p=(**char**\*) calloc(10, **sizeof**(**char**));

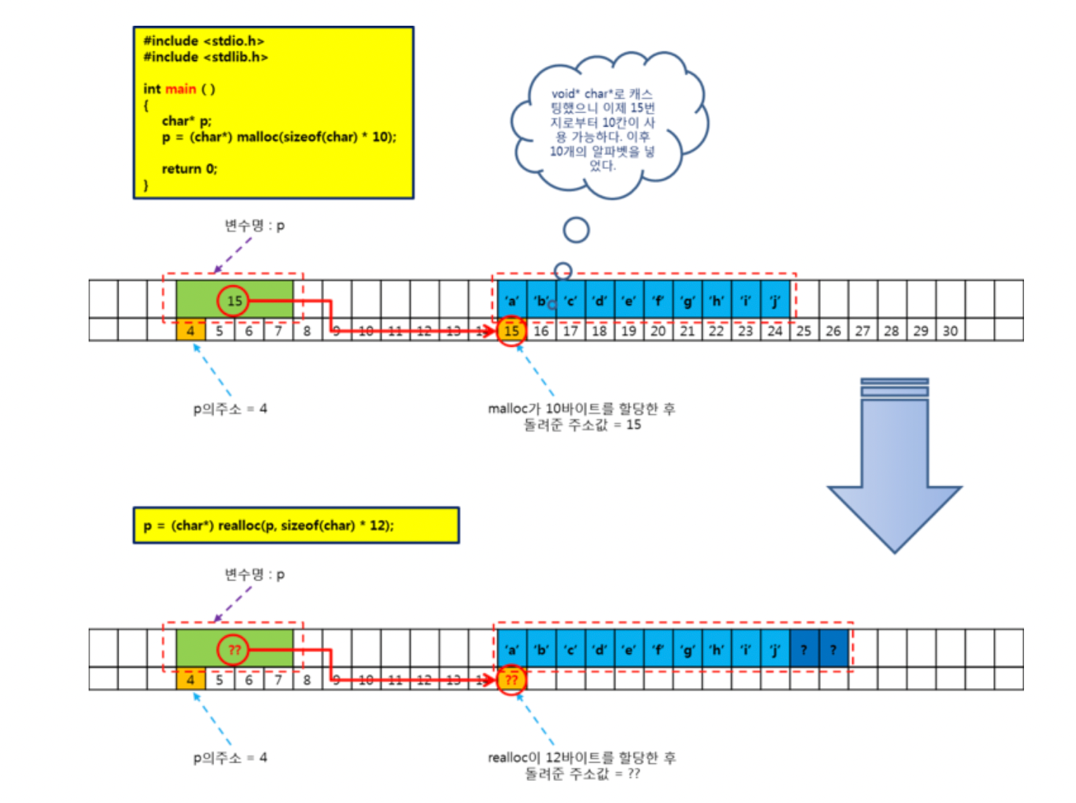
**return** 0;

}

#realloc

:malloc이나 calloc을 통해 메모리를 할당했지만 메모리 공간을 더 줄이거나 늘리기 위해 사용.

:void\* realloc (void\* p, size\_t size);



#include <stdio.h>

#include <stdlib.h>

**int** main(){

**char**\* p;

p=(**char**\*) calloc(10, **sizeof**(**char**));

p=(**char**\*) realloc(p,**sizeof**(**char**)\*12);

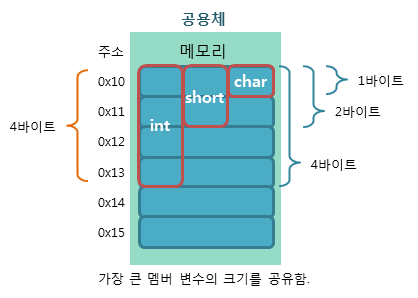
**return** 0;

}

#2 공용체(union)에 대해 조사 & 간단한 실습

공용체: 여러 멤버를 공유하는 하나의 값.

멤버 중 가장 큰 크기에 맞게 만들어짐.



#include <stdio.h>

#include <stdlib.h>

**union** a {

**int** age;

**double** grade;

};

**int** main(){

**union** a person = {20};

printf("age:%d ", person.age);

printf("avg: %lf \n",person.grade);

}

텍스트이(가) 표시된 사진

자동 생성된 설명

#3 열거형(enum)에 대해 조사

:열거형을 사용하여 정수형 상수를 편하게 정의.

:유사한 성질을 가지는 값들을 정수로 표현할 때 유용

Ex)

텍스트이(가) 표시된 사진

자동 생성된 설명

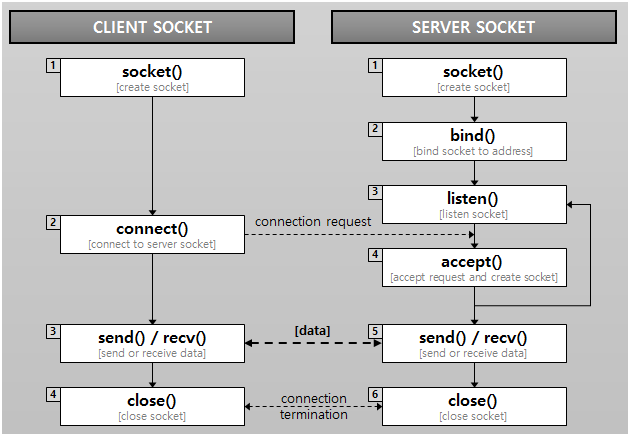
#4 Socket의 개념(종류, 역할, 용도 등 자세히)과

Socket 통신(TCP 프로토콜 통신)의 과정 조사

소켓: 프로세스가 네트워크를 통해 데이터를 주고 받고 할 수 있게 하는 창구.

: 떨어진 두 호스트를 연결해주는 인터페이스

: 서버 소켓, 클라이언트 소켓으로 나뉜다.



서버 소켓:

1. 클라이언트와 서버를 연결할 연결용 소켓 생성
2. Ip와 port 번호 결합
3. 클라이언트의 접근 요청을 수신하는 대기
4. 새로운 소켓을 만들어 클라이언트와 실질적인 연결 수행

클라이언트 소켓

1. 소켓을 생성
2. 통신할 서버에 설정된 ip와 port번호에 통신 시도
3. 서버가 accept 함수로 클라이언트의 socket descriptor 반환
4. 서로 send, recv하며 데이터 송수신

#조사과제5

텍스트이(가) 표시된 사진

자동 생성된 설명

패딩 바이트: 클래스나 구조체에 패딩 바이트를 추가하여 cpu 접근을 용이하게 해준다. 불필요한 패딩 공간을 확보하며 메모리의 크기를 맞추는 이유는 cpu의 연산 횟수를 줄이기 위해서이다.

->32비트 cpu는 한 번에 4바이트씩 읽을 수 있기 때문에 위의 경우는 5바이트지만 패딩이 추가되어서 8바이트로 나오게 되는 것이다.

# PRINT(), PRINTINFO(), swap() 함수 구현

#include<stdio.h>

typedef struct{

char Name[100];

int age;

int digit;

}Student;

void swap(Student\* a, Student\* b){

Student tmp;

tmp=\*a;

\*a=\*b;

\*b=tmp;

}

void PRINT(const Student x){

printf("Name: %s\n",x.Name);

printf("age: %d \n",x.age);

printf("digit: %d \n",x.digit); }

void PRINT\_INFO(const Student\* x){

printf("Name: %s \n",x->Name);

printf("age:%d \n",x->age);

printf("digit:%d \n",x->digit);

}

int main(){

Student a={"Miru",3,191128};

Student b={"Ruri",3,200201};

PRINT(a);

PRINT(b);

swap(&a,&b);

PRINT\_INFO(&a);

PRINT\_INFO(&b);

}

-결과

텍스트이(가) 표시된 사진

자동 생성된 설명

//친구관리프로그램

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct \_friend{

char\* name;

char\* message;

int status;

}friend;

friend\* list[100];

int list\_num;

void init();

int addfriend();

void deletefriend();

void viewfriend();

void viewall();

int main(){

friend \*frd;

list\_num=0;

int menu=0;

init();

while(1){

printf("\n");

printf("========friend menu=======");

printf("\n");

printf("1.add friend \n");

printf("2.delete friend \n");

printf("3.view friend \n");

printf("4.view all friends \n");

printf("5.exit \n");

printf("choose:");

scanf("%d",&menu);

if(menu==1)

addfriend();

if(menu==2)

deletefriend();

if(menu==3)

viewfriend();

if(menu==4)

viewall();

if(menu==5){

exit(0);

break;}

}

}

void init(){

setvbuf(stdin, 0, 2, 0);

setvbuf(stdout, 0, 2, 0);

setvbuf(stderr, 0, 2, 0);

}

int addfriend(){

int size=0;

int status;

friend \*frd=0;

frd=(friend \*)malloc(sizeof(friend));

printf("name size:");

scanf("%d", &size);

frd->name = (char \*)malloc(size);

printf("name:");

read(0, frd->name, size);

printf("message size:");

scanf("%d", &size);

frd->message = (char\*)malloc(size);

printf("message: ");

read(0, frd->message, size);

printf("status put 0(:offline) or 1(:away) or 2(:online):");

scanf("%d", &status);

list[list\_num++] = frd;

}

void deletefriend(){

int del;

printf("Choose number who to delete:");

scanf("%d",&del);

for(int i=del; i<list\_num; i++)

list[i]=list[i+1];

}

void viewfriend(){

int view;

printf("Enter index you want to view:");

scanf("%d",&view);

printf("my friend! \n");

printf("name: %s", list[view]->name);

printf("message: %s", list[view]->message);

printf("status: %s", list[view]->status);

}

void viewall(){

printf("my friend! \n");

for (int i=0; i<list\_num;i++){

printf("name: %s", list[i]->name);

printf("message: %s", list[i]->message);

printf("status: %s", list[i]->status); }

printf("\n");

}

-결과

텍스트이(가) 표시된 사진

자동 생성된 설명