



Universidad autónoma de baja california

Ingeniero en computación

Microcontroladores

Practica 1

Maestro: Jesús Adán Garcia López

Erik Garcia Chávez 01275863

viernes 7 de febrero del 2025

Desarrollo:

-Tamaño de las primitivas:

Se tienen diferentes tipos de datos primitivos los cuales tienen ciertos tamaños en memoria. Entre los que tenemos a

- **char**: valor para un carácter. tiene un tamaño que va desde los -128 a 127 bits o de 0 a 255 bits. Lo que equivale a 1 byte

- **int**: valor para definir un numero entero. tiene un tamaño que va desde los -32767 a 32768 que es equivalente a 2 bytes

-**unsigned int**: valor entero sin signo: que va desde el 0 a 65, 535 que es equivalente a 2 bytes

-**float**: tipo de dato que nos ayuda a definir un numero de tipo flotante. Tiene un tamaño de 4 bytes que equivale a 3.4×10^{-38} a $3.4 \times 10^{+38}$ bits.

-**double**: es un tipo de dato del doble de tamaño que el float, que nos ayuda a tener un numero con mas decimales después del punto flotante que es igual a 8 bytes lo que es equivalente a 1.7×10^{-308} a $1.7 \times 10^{+308}$ bits

Uso inttypes.h

Esta cabecera contiene el uso de integrales ya definidas como

-init8_t: tipo integral firmado de 8 bits

-init16_t: integral firmado de 16 bits

-init32_t: integral firmado de 32 bits

-init64_t: integral firmado de 64 bits

-unit8_t: tipo integral sin signo de 8 bits

-unit16_t: tipo integral sin signo de 16 bits

-unit32_t: tipo integral sin signo de 32 bits

-unit64_t: tipo integral sin signo de 64 bits

-initptr_t: tipo integral firmado lo suficiente grande para contener cualquier puntero

-unitptr_t: tipo integral no firmado lo suficiente grande para contener cualquier puntero

-representación de literales:

hace referencia a un valor que aparece en un programa y no se puede cambiar. en C se usa el termino constante en lugar del nombre literal. una constante literal no es direccionable lo que significa que su valor se almacena en algún lugar de la memoria. el valor literal no cambia mientras se ejecuta el programa y debe estar en el rango de valores representables para su tipo.

- Operadores Bitwise:

los operandos de los operadores bit a bit deben tener tipos enteros, pero sus tipos pueden ser diferentes, estos operadores realizan conversiones aritméticas habituales.

Tenemos:

& - and - este realiza la operación booleana de AND que comprara ambos bits, si ambos son 1 el resultado es 1 si uno de los 2 es diferente a 1 el resultado el 0

^ - OR exclusivo. Si un bit es 0 y el otro bit es 1, el bit del resultado correspondiente se establece en 1. De lo contrario, el bit del resultado correspondiente se establece en 0.

| - OR inclusivo - Si uno de los dos bits es 1, el bit del resultado correspondiente se establece en 1. De lo contrario, el bit del resultado correspondiente se establece en 0.

Se tienen las operaciones lógicas que se están en la gran mayoría de los lenguajes de programación, estos funcionan diferente, dado que las operaciones bitwise operan a nivel de los bits de las variables, comparando los bits, pero un operador lógico compara toda la variable en su forma decimal, carácter, punto flotante, etc..., son operaciones de comparación y los bitwise son más de operaciones para obtener un resultado deseado o esperado.

-apuntadores:

es una variable que contiene una dirección en memoria, la cual corresponde a la variable con la información. cada variable que cada programa necesita un espacio en memoria la cual es accesible mediante su dirección. en lenguaje C podemos trabajar con memoria usando los apuntadores. estos como su nombre lo indican apuntan a una dirección en memoria, la operación de dirección (&) regresa la dirección de una variable. el operador de indirección (*). toma la dirección de una variable y regresa el dato que contiene esa dirección.

-estructuras:

las estructuras son colecciones de variables relacionadas bajo un nombre, estas pueden contener varios tipos de datos de tipo primitivo o incluso otras estructuras, estas en lenguaje C se pueden identificar por la palabra reservada "struct". Un ejemplo de cómo declaramos una estructura en C es de la siguiente manera:

```

struct ejmp1{
    int dato1;
    float dato2;
    char carácter;
};

```

Este igual puede tener un “alias” con el cual puede ser llamado.

Unión: es un tipo de dato definido por el usuario donde todos los miembros comparten la misma ubicación de memoria. Una unión no puede contener mas de un objeto en su lista de miembros, esto significa que independientemente de cuantos miembros tiene una unión, en todo momento únicamente se usara la memoria suficiente para almacenar al miembro mas grande. Esto puede funcionar cuando se conserva memoria cuando se tienen muchos objetos y memoria limitada. Un ejemplo puede ser

```

Unión Types{
    Char ch;
    Int i;
    Long l;
    Float f;
    Double d;
};

```

Bitfield: otro tipo de dato, es una estructura que puede tener un número de bits especificado. Denominado campo de bits – bitfield. Su longitud se establece en dos puntos desde el declarador para el nombre del campo. En campo de bits se interpreta como un tipo entero

Struct-declarator:

Declarador

Type-specifier declarador opt: constant-expression

Constant-expression: especifica el ancho del campo en bits.

Type-specifier: debe ser **unsigned int**, **signed int** o **int**.

-palabras reservadas:

las palabras reservadas son palabras que tiene un significado importante en el compilador de C. el lenguaje C posee un numero reducido de palabras reservada que define el standard ANSI-C. las cuales son los nombres de los tipos de datos primitivos, tipo de dato de una función, estructuras, ciclos, etc.

-Condicionales del pre-compilador

las líneas que se indican con un (#) estas se indica en la primera columna se llaman líneas de control y se comunican con el procesador. el efecto de una línea de control abarca desde su lugar en el archivo hasta el final del mismo.

pongamos por ejemplo la directiva " #include " esto ocasiona que el compilador incluya un archivo fuente. este archivo estará entre comillas (" ") o entra ángulos (<>). dichos archivos reciben el nombre de archivos cabecera (head) por eso se usa la extensión.h.

-casteo:

el casting nos va a permitir hacer una conversión explicita de un tipo de dato a otro, este se hace a través de un operador de conversión de tipo "type casting operator" en C se entiende como en Cast tradicional el cual es mediante un operador de conversión de tipo que es poner antes el tipo de dato al que se quiere cambiar y después la variable.

-GitHub:

GitHub es una plataforma de control de versiones. Este permite a los desarrolladores colaboración y realizar cambios en proyectos compartidos, un control de versiones es un sistema que ayuda a rastrear y gestionar los cambios realizados en un archivo o un conjunto de estos. Esto permite hace cambios al código tanto como sea necesario y si alguna versión resulta estar corrompida esta puede ser desechada volviendo a un commit atrás.

-conclusión:

Durante el desarrollo de la practica investigue conceptos que ya conocía, no había profundizado tanto como las condiciones pre-compilador, que es algo que nos dice que va, pero no el porqué, en la materia será algo muy importante ya que nosotros estaremos realizando nuestras librerías por lo que el saber cómo llamarlas va ser tema fundamental. Y otros temas como el casteo que no sabía que se podría hacer en C lo conocía en

Python y en java con las clases, pero fue algo nuevo que aprendí, por lo que puede conocer mas sobre el lenguaje y la amplia gama de utilidades que tiene pese al tiempo que lleva en el mercado.

Bibliografía:

<https://www.ibm.com/docs/es/aix/7.3?topic=files-inttypesh-file>

<https://www.ibm.com/docs/es/i/7.5?topic=tokens-literals>

<https://learn.microsoft.com/es-es/cpp/c-language/c-bitwise-operators?view=msvc-170>

<https://www.fing.edu.uy/tecnoinf/mvd/cursos/prinprog/material/teo/prinprog-teorico08.pdf>

<https://www.ibm.com/docs/es/developer-for-zos/14.2?topic=programs-c-reserved-keywords>

http://www.sc.ehu.es/sbweb/webcentro/automatica/web_avr/archivos/Ensamblador%20AVRs/directivas/direc_preprocesador_tipo_C.htm

<http://www.itlalaguna.edu.mx/2014/Oferta%20Educativa/Ingenierias/Sistemas/Plan%201997-2004/programacion2/cpp4.pdf>

<https://programacionnerd.blogspot.com/2012/05/c-conversiones-de-tipo-casting-en-c.html>

<https://learn.microsoft.com/es-es/cpp/cpp/unions?view=msvc-170>

<https://learn.microsoft.com/es-es/cpp/c-language/c-bit-fields?view=msvc-170>