Ejercicios Propuestos - Concepts of Programming Languages

13 de febrero de 2014

1. Capítulo 6: Expressions and assignment statements

1.1. pregunta 1

Design a set of simple test programs to determine the type compatibility rules of a C compiler to which you have access. Write a report of your findings. La importancia de saber manejar la compatibilidad entre los tipos de datos, es poder modificar los datos sin la necesidad de realizar una nueva asignación

```
#include <stdlib.h>
#include <stdio.h>
int main (int argc, char * argv[]) {
      int w = 23;
      long int x= 3423232,s;

      s = w;
      w = x;

      printf("El valor de s es= %ld \n Y w=%d", s, w);
      return 0;
}
```

En este ejemplo se obtuvo como resultado que s era igual a 23, mientras que w obtuvo un valor totalmente diferente. Debido a que$w$ era un tipo de dato entero. Se origino el problema de incompatibilidad.
La forma en como los tipos de compatibilidad se comportan, varian de acuerdo a ciertos casos.

La mezcla entre la compilacion y como eso afectan en las operaciones. Por ejemplo en el caso del int no causa error en el hecho de haber intentando usar un tipo de dato long. Sin embargo las variables pueden usarse para anticipar una asignacion o el traspaso de parametros.

1.2. pregunta 2

Determine whether some C compiler to which you have access implements the free function.

```
#include<stdio.h>

        int main()
        {
                int *ptr_one;

                ptr_one = (int *)malloc(sizeof(int));

                if (ptr_one == 0)
                {
                        printf("ERROR: Out of memory\n");
                        return 1;
                }

                *ptr_one = 25;
                printf("%d\n", *ptr_one);

                free(ptr_one);

                return 0;
        }
```

1.3. Pregunta 7

Write a C program that does a large number of references to elements of two-dimensioned arrays, using only subscripting. Write a second program that does the same operations but uses pointers and pointer arithmetic for the storage-mapping function to do the array references.Compare the time efficiency of the two pro-

grams. Which of the two programs is likely to be more reliable? Why?

```
#define L 100
#define M 100
int a[L][M]

a[i] = *(a + i)
```

Con respecto a los tiempos de ejecucion, mas rapido es el primer caso, ya que la lectura de los datos es directamente a la memoria proporcionada por el registro a. Mientras que en el otro caso, se debe acceder primero a la memoria, buscar la direccion en la que se encuentra almacenado y ahi realizar la asignacion.