

Soluciones 1er Parcial Informática I

Ing. Ignacio Bonelli, Ing. Héctor Levi, Leandro Kollenberger

Sábado 11 de Julio de 2015

Preguntas

Parte Teórica

1. Defina dirección y contenido de una variable.
2. ¿Cuál es la diferencia entre un archivo .o y un archivo .c? Ejemplo: entradaTeatro.o y entradaTeatro.c.
3. ¿Qué tipo de sistema de numeración se utiliza para los permisos de Linux?
4. Dadas las siguientes declaraciones:

```
int v[4] = {50, 34, 15, 45};
char w = 'a';
float z[3] = {18.1, 24, 20.4};
```

Si asumimos que el sistema operativo asignará en forma sucesiva de direcciones de memoria a partir de `0xafff0000` y enteros de 2bytes:

- (a) ¿Qué dirección de memoria contendrá el valor `v[1]`?
 - (b) ¿Qué dirección de memoria resultará de resolver `v+2`?
 - (c) ¿En qué dirección de memoria estará el caracter `'a'`?
 - (d) ¿Qué dirección de memoria contendrá el valor `z[0]`?
 - (e) ¿Cuál es el contenido de la dirección `0xafff0008`?
 - (f) ¿Cuál es el contenido de la dirección `0xafff000D`?
5. (a) ¿Cuál es la diferencia entre un header y una librería?
 - (b) ¿Qué contiene cada uno?
 - (c) ¿Dónde especificamos la inclusión de un header en nuestro programa?
 - (d) ¿Dónde y cómo especificamos la inclusión de una librería en nuestro programa?

Parte Práctica

1. Se tiene un sistema que mide experimentos y mide la fuerza como un vector con dirección e intensidad. Debemos realizar un programa que procese estos experimentos y devuelva un valor medio de las series de experimentos. Lo que siguen son las diferentes partes de ese programa general.
 - (a) Escribir una función que recibe varios nombres de archivos via argumentos del main. Leer los nombres de los archivos e invocar la función `leerarchivo()` pasando como argumento el nombre del archivo. Realizar tantos llamados como nombres haya recibido el programa.
 - (b) Escribir una función que recibe: Un nombre de archivo y un puntero a un vector. El archivo contiene un par de números por línea. Estos números son dos enteros separados por coma (todo en ASCII), y el archivo tiene como máximo 10 líneas. Leer las líneas en un vector de 2 dimensiones y devolver la cantidad de líneas que se leyeron.
 - (c) Realizar un función que recibe: Un puntero a un vector de dos dimensiones, un entero (máximo 10) y un puntero a un vector de dos flotantes (valor a devolver). El vector tendrá 2 posiciones ('x' e 'y') por "línea" y la cantidad de pares/líneas definida por el segundo entero. Cada posición es un vector con origen 0,0 y tamaño x,y. Realizar el promedio de todos los vectores de ese conjunto y devolverlo dentro del puntero a flotantes pasado como argumento en el llamado a la función.
2. Realizar una función que reciba una cadena de caracteres en ASCII de hasta 8192 caracteres. La función deberá leer la cadena y determinar (ignorando la diferencia entre mayúsculas y minúsculas) la cantidad de veces que se repite cada carácter del abecedario. El programa terminará mostrando por pantalla (de la A a la Z sin la Ñ) qué cantidad de ocurrencias de cada letra se encontró dentro de la cadena de caracteres recibida.

Respuestas

Parte Teórica

1. La dirección de una variable es su posición en la memoria de la computadora, y su contenido es su valor numérico.
2. Un archivo `.c` incluye el código fuente del programa en lenguaje C, y los archivos `.o` (archivos objeto) incluyen el código máquina (binario) compilado del programa.
3. En linux los permisos se expresan en numeración octal, por ejemplo de la siguiente manera:

6	4	5
↓	↓	↓
110	100	101
<i>rw</i> <i>x</i>	<i>rw</i> <i>x</i>	<i>rw</i> <i>x</i>
<i>usuario</i>	<i>grupo</i>	<i>otros</i>

4. Mapa de memoria:

0xAFFF 0000	v[0]	(a) 0xAFFF 0002
0xAFFF 0002	v[1]	(b) 0xAFFF 0004
0xAFFF 0004	v[2]	(c) 0xAFFF 0008
0xAFFF 0006	v[3]	(d) 0xAFFF 0009
0xAFFF 0008	w	(e) w → 'a'.
0xAFFF 0009	z[0]	(f) z[1] → 24.
0xAFFF 000D		
0xAFFF 0011	z[1]	
0xAFFF 0015	z[2]	

5. (a) Un archivo header (cabecera) es un archivo de texto en lenguaje C, mientras que una librería es un archivo binario en código máquina.
- (b) El header contiene los prototipos de las funciones, mientras que las librerías contienen las funciones desarrolladas, compiladas y listas para ser linkeadas con nuestro programa.
- (c) Se especifica al comienzo del archivo con la directiva del preprocesador de C `#include`.
- (d) Se la especifica con la directiva `-l(libreria)` al linker, por ejemplo `-lpthread` para `libpthread`, o `-lm` para `libm`, la librería matemática.

Parte Práctica

1. (a)

```

void parseargs(int argc, char *argv[]) {
    int i;
    if(argc < 2) return;
    for(i = 1; i < argc; i++)
        leerarchivo(argv[i]);
    return;
}

```
 - (b)

```

int loadvec(char *filename, int *vec[]) {
    int i;
    FILE *f;
    f = fopen(filename, "r");
    if(f == NULL) return -1;
    while(!feof(f)) {
        for(i = 0; i < 10; i++)
            fscanf(f, "%d,%d\n", &vec[i][0], &vec[i][1]);
    }
    return i;
}

```
 - (c)

```

void averagevec(int *vec[], int n, float *result) {
    int i;
    result[0] = 0; result[1] = 0;
    for(i = 0; i < n; i++) {
        result[0] += (float) vec[i][0];
        result[1] += (float) vec[i][1];
    }
    result[0] /= n;
    result[1] /= n;
}

```
2.

```

void printchars(char *str) {
    int letters[26];
    int len = strlen(str);
    int i;

    for(i = 0; i < len; i++) {
        if( (str[i] >= 'A') && (str[i] <= 'Z') ) //mayuscula
            letters[str[i] - 'A']++;
        if( (str[i] >= 'a') && (str[i] <= 'z') ) //minuscula
            letters[str[i] - 'a']++;
    }

    for(i = 0; i < 26; i++)
        printf("Cantidad de %c: %d\n", (char) (i + 'A'), letters[i]);

    return;
}

```