

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

PRUEBAS DE SOFTWARE

Nombre: Alisson Clavijo

NRC: 9870

1.- Especificación del programa “count” obtenida a partir del código

El comportamiento del programa obtenido a partir del código es:

Todos los argumentos de la línea de comandos se tratan como nombres de ficheros, aunque siempre se intenta leer uno de más que no existe. Se distinguen tres casos:

Se proporcionan argumentos, pero no hay ficheros con nombres correspondientes a los argumentos. En este caso, el programa se para con un mensaje de error que sale por la salida estándar. Se proporcionan argumentos, y corresponden a ficheros que existen. Para cada fichero, el número de líneas, palabras y caracteres se cuenta y se imprime la suma junto con el nombre del programa. No se proporcionan argumentos. En este caso, el programa intenta leer de algo que no está inicializado y procede como en (2), a pesar de que no se imprime el nombre del programa. Si el número total de argumentos es al menos 1, se imprime la suma total más uno de todos los caracteres y palabras en todos los ficheros y el número de líneas del último fichero.

2.- Especificación original del programa “count

count cuenta el número de líneas, palabras y caracteres que hay en cada fichero que se le pasa como argumento. Las palabras son secuencias de caracteres que están separadas por uno o más espacios, tabuladores o saltos de línea. Si alguno de los ficheros que se le pasa como argumento no existe, aparece por la salida de error el mensaje de error correspondiente y se continúa procesando el resto de ficheros. Si no se indica ningún fichero, **count** lee de la entrada estándar. Se muestra por pantalla los valores computados para cada fichero (junto con el nombre del fichero) así como la suma total de cada valor para todos los ficheros. Si se procesa un único fichero o si se procesan datos provenientes de la entrada estándar, entonces no se imprime ninguna suma. La salida se imprime en el orden siguiente: primero líneas, luego palabras, luego caracteres y finalmente bien el nombre del fichero o la palabra “total” para la suma. Si se ha leído de la entrada estándar, el cuarto valor (nombre) no aparece

3.-Comparación de ambas especificaciones y detección de defectos.

Se señalan en negrita los puntos donde la especificación obtenida por abstracción diverge de la especificación original:

No.	Especificación Original	Especificación por abstracción	Tipo de defecto	Inconformidad
1	Los ficheros que se le pasa como argumento no existe, aparece por la salida de error el mensaje de error correspondiente y se continúa procesando el resto de ficheros.	Se proporcionan argumentos, pero no hay ficheros con nombres correspondientes a los argumentos. En este caso, el programa se para con un mensaje de error que sale por la salida estándar.	Incorrecto	En la especificación original nos dice que sale un mensaje de error y continua el proceso, y en la especificación por abstracción se interpreta que nos da un mensaje de error y se para el proceso.

2	La salida se imprime en el orden siguiente: primero líneas, luego palabras, luego caracteres y finalmente bien el nombre del fichero o la palabra “total” para la suma.	Se proporcionan argumentos, y corresponden a ficheros que existen. Para cada fichero, el número de líneas, palabras y caracteres se cuenta y se imprime la suma junto con el nombre del programa.	Incorrecto	En la especificación original nos da un orden para imprimir y nos dice que se va a imprimir el nombre del fichero o la palabra total y en la especificación por abstracción con el código se interpreta que se va a imprimir la suma junto al nombre del programa y no tiene un orden y los parámetros línea, palabras, caracteres.
3	entonces no se imprime ninguna suma. La salida se imprime en el orden siguiente: primero líneas, luego palabras, luego caracteres	se imprime la suma total más uno de todos los caracteres y palabras en todos los ficheros	Incorrecto	En la especificación original nos dice que no se imprime ninguna suma, y en la especificación por abstracción, en el código vemos como se imprime la suma total mas uno.
líneas 9 -10	count cuenta el número de líneas, palabras y caracteres que hay en cada fichero que se le pasa como argumento.	Se declaran y se les asigna el valor inicial de 1 a tres variables de tipo long: tlinect, twordct y tcharct	Incorrecto	Las variables están inicializadas a 1 y deberían estar a 0
Línea 14	argc > 1 → (file-open(argv[i]) = fallo → msgerr y parar fp:= stream) ()	Se utiliza una estructura condicional if para verificar si argc (el número de argumentos en línea de comandos) es mayor que 1 y si el archivo argv[i] se puede abrir. Si ambas condiciones se cumplen, el bloque de código dentro del if se ejecuta. Dentro del bloque if, se utiliza la función fopen para abrir el archivo argv[i] en modo lectura ("r"). Si fopen devuelve un puntero nulo, significa que no se pudo abrir el archivo	Incorrecto	La variable “fp” no está inicializada en caso de que la entrada se tome de “stdin” (entrada estándar).
Línea 16	El componente termina con “exit(1)”.	s e inicializan las variables linect, wordct, charct e inword a 0.	Incorrecto	El componente termina con “exit(1)”, donde debería haberse usado un “continue”
Línea 31	Se usa *argv.	si argc no es mayor que 1, se imprime una nueva línea	Incorrecto	Se usa *argv en lugar de argv[i]

4.- Faltas en las líneas de código de “contador.c”.

Señale las faltas en el código que se generan por los defectos encontrados en la abstracción sucesiva. (marcar con resaltador). OJO el defecto de código debe tener relación con el defecto en la especificación.

```
#include <stdio.h>
main (argc, argv)
    int argc;
    char *argv[];
{
    int c, i, inword;
    FILE *fp;
    long linect, wordct, charct;
    long tlinect = 1, twordct = 1, tcharct = 1;
    i = 1;
    do {
        if (argc > 1 && (fp=fopen(argv[i], "r")) == NULL) {
            fprintf (stdout, "can't open %s\n", argv[i]);
            exit (1);
        }
        linect = wordct = charct = 0;
        inword = 0;
        while ((c = getc(fp)) != EOF) {
            ++charct;
            if (c == '\n')
                ++linect;
            if (c == ' ' || c == '\t' || c == '\n')
                inword = 0;
            else if (inword == 0) {
                inword = 1;
                ++wordct;
            }
        }
        printf("%7ld %7ld %7ld", linect, wordct, charct);
        if (argc > 1)
            printf(" %s\n", *argv);
        else
            printf("\n");
        fclose(fp);
        tlinect += linect;
        twordct += wordct;
        tcharct += charct;
    } while (++i <= argc);
    if (argc > 1)
        printf("%7ld %7ld %7ld total\n", linect, twordct, tcharct);
    exit(0);
}
```