

Trabajo Práctico 0

De Rosa - Schapira - Guerrero

Segundo Cuatrimestre 2017

Índice

1. Documentación relevante al diseño e implementación del programa	1
1.1. Objetivos	1
1.2. Problema	1
1.3. Implementación	1
1.3.1. Entorno	1
1.3.2. Complicaciones	1
1.3.3. Programación	1
2. Código	2
2.1. Comandos para compilar el programa	2
2.2. Código fuente en lenguaje c	2
3. Código MIPS32 generado por el compilador	6
4. Corridas de prueba	25
4.1. Primer corrida	25
4.2. Segunda corrida	25
4.3. Tercera corrida	25

1. Documentación relevante al diseño e implementación del programa

1.1. Objetivos

Familiarizarse con las herramientas de software que usarán a lo largo de la cursada.

1.2. Problema

Se busca implementar un programa que busque las palabras capicúas dentro de una entrada determinada. Por capicúa se considera a toda combinación de caracteres validos que pueda ser leida de igual maneara de izquierda a derecha como de derecha a izquierda. Bajo este contexto, se consideran a los siguientes caracteres ASCII como validos:

- a-z
- 0-9
- Signo Menos
- Guión Bajo

Mientras que el resto de los caracteres no mencionados se toman como espacios entre palabras.

1.3. Implementación

1.3.1. Entorno

Utilizando el programa GXemul, se simula el entorno de desarrollo que acompañara en futuros trabajos junto con una máquina MIPS corriendo una versión reciente del sistema operativo NetBSD.

1.3.2. Complicaciones

El entorno limitado en cuestión impide el uso de determinadas librerías que simplificarían la implementación del programa agregando una complicación extra al desarrollo.

Por otro lado se presenta el desafío de utilizar herramientas nuevas siendo estas GXemul junto a NetBSD. El networking de GXemul no presenta una interfaz de red “real” en el host OS, sino que abre sockets TCP/IP. Como consecuencia, no se pueden realizar conexiones entrantes de forma directa al guest OS requiriendo así de la creación de un Túnel SSH hacia el guest OS.

1.3.3. Programación

Para la implementación del problema se utilizara el lenguaje de programación C.

2. Código

2.1. Comandos para compilar el programa

Nos paramos en el directorio donde tenemos los archivos y ejecutamos:
gcc -Wall tp.c

2.2. Código fuente en lenguaje c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <unistd.h>
#include <getopt.h>

#define ARRAY_SIZE 1024
#define INPUT_SIZE 1024

/* Tipo de argumentos esperados */
struct arguments{
    char* output_file;          /* file arg to '--output' */
    char* input_file;           /* file arg to '--input' */
};

void print_help(){
    printf("Usage:\n \
\ttp0 -h\n \
\ttp0 -v\n \
\ttp0 [options]\nOptions:\n \
\t-v, --version      \tPrint version and quit.\n \
\t-h, --help         \tPrint help and quit.\n \
\t-i, --input         \tPath to input file.\n \
\t-o, --output        \tPath to output file.\nExamples:\n \
\ttp0 -i ~/input -o ~/output\n");
}

void print_version(){
    printf("tp0 [Guerrero - Schapira - De Rosa] 0.5\n");
}

/* Funcion que invierte una cadena de caracteres */
char* strrev(char *str){
    char *p1, *p2;

    if (! str || ! *str) return str;
```

```

        for (p1 = str, p2 = str + strlen(str) - 1; p2 > p1; ++p1, --p2){
            *p1 ^= *p2;
            *p2 ^= *p1;
            *p1 ^= *p2;
        }
        return str;
    }

/*Se hace una copia de la palabra recibida, se la invierte y se verifica
si son iguales o no */
int es_capicua(char* string){
    /*Pasamos a minuscula los caracteres*/
    char lower[strlen(string)];
    memset(lower, 0, sizeof(lower));

    int i;
    for(i = 0; i <= strlen(string); i++){
        lower[i] = tolower(string[i]);
    }
    /*Verificamos si es capicua*/
    char copy[strlen(lower)];
    strcpy(copy, lower);
    strrev(copy);
    if(!strcmp(lower, copy) && strlen(lower) > 1 ) return 1;
    return 0;
}

/*Chequea que palabras de la linea recibida son capicua*/
int palabras_en_linea(char* string, char* array_strings){
    int len = strlen(string);
    int init = 0;
    int cant = 0;
    int words = 0;
    char new[100];

    int i, j;
    for(i = 0; i <= len; i++){
        int a = string[i];
        if( a>47 && a<58 ) continue;
        if( (a>64 && a<91) || (a > 96 && a<123) ) continue;
        if(a == 95 || a==45) continue;

        //Si estamos aca es porque es un caracter que quivale al espacio.
        memset(new, 0, sizeof(new));

        for(j = 0; j < (i - init); j++){

```

```

        new[j] = string[init + j];
    }

    if(es_capicua(new)){
        for(j = 0; j < (i - init); j++){
            array_strings[cant] = new[j];
            cant++;
        }
        array_strings[cant] = '\n';
        cant++;
        words++;
    }
    init = i + 1;
}

    return words;
}

int main(int argc, char* argv[]){
    /* Lectura de entrada */
    /* Default values. */
    struct arguments arguments;
    arguments.output_file = NULL;
    arguments.input_file = NULL;

    int c;
    while (1){
        static struct option long_options[] =
        {
            /* These options dont set a flag.
               We distinguish them by their indices. */
            {"help", no_argument, 0, 'h'},
            {"version", no_argument, 0, 'v'},
            {"input", required_argument, 0, 'i'},
            {"output", required_argument, 0, 'o'},
            {0, 0, 0, 0}
        };
        /* getopt_long stores the option index here. */
        int option_index = 0;

        c = getopt_long (argc, argv, "hvc:i:o:", long_options, &option_index);

        /* Detect the end of the options. */
        if (c == -1) break;

        switch (c){

```

```

    case 0:
        /* If this option set a flag, do nothing else now. */
        if (long_options[option_index].flag != 0) break;
        printf ("option %s", long_options[option_index].name);
        if (optarg) printf (" with arg %s", optarg);
        printf ("\n");
        break;
    case 'h':
        print_help();
        exit(0);
        break;
    case 'v':
        print_version();
        exit(0);
        break;
    case 'i':
        arguments.input_file = optarg;
        break;
    case 'o':
        arguments.output_file = optarg;
        break;
    case '?':
        break;
    default:
        abort ();
}
}
/* Print any remaining command line arguments (not options). */
if (optind < argc){
    printf ("non-option ARGV-elements: ");
    while (optind < argc) printf ("%s ", argv[optind++]);
    putchar ('\n');
}

/*Procesamiento de palabra capicua*/
char input[INPUT_SIZE];
char array[ARRAY_SIZE];
memset(array, 0, sizeof(array));

FILE* entrada = fopen(arguments.input_file, "r");

//Si no hay archivo, pedimos por teclado.
if(!entrada){
    printf("Ingrese la oracion a evaluar:\n");
    fgets(input, INPUT_SIZE, stdin);
}

```

```

        palabras_en_linea(input , array );
    }

//Si hay, procesamos las lineas.
else{
    while( !feof(entrada) ){
        char ent[INPUT_SIZE];
        memset(ent, 0, sizeof(ent));
        /*Cargo una linea*/
        fgets(ent, sizeof(ent), entrada);

        if(ent[0] == '\0'){
            continue;
        }

        /*Proceso una linea*/
        char aux[INPUT_SIZE];
        memset(aux, 0, sizeof(aux));

        palabras_en_linea(ent, aux);
        strcat(array, aux);
    }
    fclose(entrada);
}

/*MANEJAMOS LA SALIDA*/

FILE* salida = fopen(arguments.output_file , "w");

//Si no hay archivo, imprimimos por pantalla
if(!salida){
    printf("Palabras capicua:\n%s", array);
}

//Si hay archivo, lo guardamos en el
else{
    fputs( array, salida);
    fclose(salida);
}

return 0;
}

```

3. Código MIPS32 generado por el compilador

```
.file 1 "tp.c"
```



```

.section .mdebug.abi32
.previous
.abicalls
.rdata
.align 2

$LC0:
.ascii "Usage:\n"
.ascii "    \ttp0 -h\n"
.ascii "    \ttp0 -v\n"
.ascii "    \ttp0 [options]\n"
.ascii "Options:\n"
.ascii "    \t-v, --version    \tPrint version and quit.\n"
.ascii "    \t-h, --help       \tPrint help and quit.\n"
.ascii "    \t-i, --input      \tPath to input file.\n"
.ascii "    \t-o, --output     \tPath to output file.\n"
.ascii "Examples:\n"
.ascii "    \ttp0 -i ~/input -o ~/output\n\000"
.text
.align 2
.globl print_help
.ent    print_help

print_help:
.frame $fp,40,$ra          # vars= 0, regs= 3/0, args= 16, extra= 0
.mask  0xd0000000,-8
.fmask 0x00000000,0
.set    noreorder
.cpload $t9
.set    reorder
subu    $sp,$sp,40
.cprestore 16
sw      $ra,32($sp)
sw      $fp,28($sp)
sw      $gp,24($sp)
move    $fp,$sp
la      $a0,$LC0
la      $t9,printf
jal     $ra,$t9
move    $sp,$fp
lw      $ra,32($sp)
lw      $fp,28($sp)
addu    $sp,$sp,40
j       $ra
.end     print_help
.size   print_help,.-print_help
.rdata
.align 2

```

```

$LC1:
    .ascii    "tp0 [Guerrero - Schapira - De Rosa] 0.5\n\000"
    .text
    .align    2
    .globl    print_version
    .ent      print_version

print_version:
    .frame    $fp,40,$ra                                # vars= 0, regs= 3/0, args= 16, extra= 0
    .mask     0xd0000000,-8
    .fmask    0x00000000,0
    .set      noreorder
    .cpload   $t9
    .set      reorder
    subu      $sp,$sp,40
    .cprestore 16
    sw        $ra,32($sp)
    sw        $fp,28($sp)
    sw        $gp,24($sp)
    move      $fp,$sp
    la        $a0,$LC1
    la        $t9,printf
    jal       $ra,$t9
    move      $sp,$fp
    lw        $ra,32($sp)
    lw        $fp,28($sp)
    addu      $sp,$sp,40
    j         $ra
    .end      print_version
    .size     print_version,.-print_version
    .align    2
    .globl    strrev
    .ent      strrev

strrev:
    .frame    $fp,56,$ra                                # vars= 16, regs= 3/0, args= 16, extra= 0
    .mask     0xd0000000,-8
    .fmask    0x00000000,0
    .set      noreorder
    .cpload   $t9
    .set      reorder
    subu      $sp,$sp,56
    .cprestore 16
    sw        $ra,48($sp)
    sw        $fp,44($sp)
    sw        $gp,40($sp)
    move      $fp,$sp
    sw        $a0,56($fp)

```

```

        lw      $v0,56($fp)
        beq     $v0,$zero,$L21
        lw      $v0,56($fp)
        lb      $v0,0($v0)
        bne     $v0,$zero,$L20
$L21:
        lw      $v0,56($fp)
        sw      $v0,32($fp)
        b       $L19
$L20:
        lw      $v0,56($fp)
        sw      $v0,24($fp)
        lw      $a0,56($fp)
        la      $t9,strlen
        jal     $ra,$t9
        move    $v1,$v0
        lw      $v0,56($fp)
        addu    $v0,$v1,$v0
        addu    $v0,$v0,-1
        sw      $v0,28($fp)
$L22:
        lw      $v0,28($fp)
        lw      $v1,24($fp)
        sltu    $v0,$v1,$v0
        bne     $v0,$zero,$L25
        b       $L23
$L25:
        lw      $a1,24($fp)
        lw      $v0,24($fp)
        lw      $v1,28($fp)
        lbu     $a0,0($v0)
        lbu     $v0,0($v1)
        xor     $v0,$a0,$v0
        sb      $v0,0($a1)
        lw      $a1,28($fp)
        lw      $v0,28($fp)
        lw      $v1,24($fp)
        lbu     $a0,0($v0)
        lbu     $v0,0($v1)
        xor     $v0,$a0,$v0
        sb      $v0,0($a1)
        lw      $a1,24($fp)
        lw      $v0,24($fp)
        lw      $v1,28($fp)
        lbu     $a0,0($v0)
        lbu     $v0,0($v1)

```

```

        xor    $v0,$a0,$v0
        sb     $v0,0($a1)
        lw     $v0,24($fp)
        addu   $v0,$v0,1
        sw     $v0,24($fp)
        lw     $v0,28($fp)
        addu   $v0,$v0,-1
        sw     $v0,28($fp)
        b      $L22
$L23:
        lw     $v0,56($fp)
        sw     $v0,32($fp)
$L19:
        lw     $v0,32($fp)
        move   $sp,$fp
        lw     $ra,48($sp)
        lw     $fp,44($sp)
        addu   $sp,$sp,56
        j      $ra
        .end   strrev
        .size   strrev,.-strrev
        .align  2
        .globl  es_capicua
        .ent    es_capicua
es_capicua:
        .frame  $fp,56,$ra
        .mask   0xd0010000,-4
        .fmask  0x00000000,0
        .set    noreorder
        .cload  $t9
        .set    reorder
        subu    $sp,$sp,56
        .cprestore 16
        sw     $ra,52($sp)
        sw     $fp,48($sp)
        sw     $gp,44($sp)
        sw     $s0,40($sp)
        move   $fp,$sp
        sw     $a0,56($fp)
        sw     $sp,32($fp)
        lw     $a0,56($fp)
        la     $t9,strlen
        jal    $ra,$t9
        move   $v1,$v0
        addu   $v1,$v1,-1
        addu   $v0,$v1,1

```

vars= 16, regs= 4/0, args= 16, extra= 0

```

        addu    $v0,$v0,7
        srl     $v0,$v0,3
        sll     $v0,$v0,3
        subu    $sp,$sp,$v0
        addu    $v0,$sp,16
        sw      $v0,36($fp)
        addu    $v0,$v1,1
        lw      $a0,36($fp)
        move    $a1,$zero
        move    $a2,$v0
        la      $t9,memset
        jal     $ra,$t9
        sw      $zero,24($fp)
$L27:
        lw      $a0,56($fp)
        la      $t9,strlen
        jal     $ra,$t9
        lw      $v1,24($fp)
        sltu    $v0,$v0,$v1
        beq     $v0,$zero,$L30
        b       $L28
$L30:
        lw      $v0,24($fp)
        lw      $v1,36($fp)
        addu    $a0,$v1,$v0
        lw      $v1,56($fp)
        lw      $v0,24($fp)
        addu    $v0,$v1,$v0
        lb      $v0,0($v0)
        sll     $v1,$v0,1
        lw      $v0,_tolower_tab_
        addu    $v0,$v1,$v0
        addu    $v0,$v0,2
        lbu     $v0,0($v0)
        sb      $v0,0($a0)
        lw      $v0,24($fp)
        addu    $v0,$v0,1
        sw      $v0,24($fp)
        b       $L27
$L28:
        lw      $a0,36($fp)
        la      $t9,strlen
        jal     $ra,$t9
        addu    $v0,$v0,-1
        addu    $v0,$v0,1
        addu    $v0,$v0,7

```

```

        srl      $v0,$v0,3
        sll      $v0,$v0,3
        subu     $sp,$sp,$v0
        addu     $s0,$sp,16
        move     $a0,$s0
        lw       $a1,36($fp)
        la       $t9,strcpy
        jal      $ra,$t9
        move     $a0,$s0
        la       $t9,strrev
        jal      $ra,$t9
        lw       $a0,36($fp)
        move     $a1,$s0
        la       $t9,strcmp
        jal      $ra,$t9
        bne      $v0,$zero,$L31
        lw       $a0,36($fp)
        la       $t9,strlen
        jal      $ra,$t9
        sltu     $v0,$v0,2
        bne      $v0,$zero,$L31
        lw       $sp,32($fp)
        li       $v0,1                                # 0x1
        sw       $v0,28($fp)
        b        $L26
$L31:
        lw       $sp,32($fp)
        sw       $zero,28($fp)
$L26:
        lw       $v0,28($fp)
        move     $sp,$fp
        lw       $ra,52($sp)
        lw       $fp,48($sp)
        lw       $s0,40($sp)
        addu     $sp,$sp,56
        j        $ra
        .end     es_capicua
        .size    es_capicua,.-es_capicua
        .align   2
        .globl   palabras_en_linea
        .ent     palabras_en_linea
palabras_en_linea:
        .frame   $fp,176,$ra                        # vars= 136, regs= 3/0, args= 16, ext
        .mask    0xd0000000,-8
        .fmask   0x00000000,0
        .set     noreorder

```

```

        .cpload $t9
        .set     reorder
        subu     $sp,$sp,176
        .cprestore 16
        sw       $ra,168($sp)
        sw       $fp,164($sp)
        sw       $gp,160($sp)
        move     $fp,$sp
        sw       $a0,176($fp)
        sw       $a1,180($fp)
        lw       $a0,176($fp)
        la       $t9,strlen
        jal      $ra,$t9
        sw       $v0,24($fp)
        sw       $zero,28($fp)
        sw       $zero,32($fp)
        sw       $zero,36($fp)
        sw       $zero,144($fp)
$L33:
        lw       $v0,144($fp)
        lw       $v1,24($fp)
        slt      $v0,$v1,$v0
        beq      $v0,$zero,$L36
        b        $L34
$L36:
        lw       $v1,176($fp)
        lw       $v0,144($fp)
        addu     $v0,$v1,$v0
        lb       $v0,0($v0)
        sw       $v0,152($fp)
        lw       $v0,152($fp)
        slt      $v0,$v0,48
        bne     $v0,$zero,$L37
        lw       $v0,152($fp)
        slt      $v0,$v0,58
        beq      $v0,$zero,$L37
        b        $L35
$L37:
        lw       $v0,152($fp)
        slt      $v0,$v0,65
        bne     $v0,$zero,$L40
        lw       $v0,152($fp)
        slt      $v0,$v0,91
        bne     $v0,$zero,$L35
$L40:
        lw       $v0,152($fp)

```

```

        slt      $v0,$v0,97
        bne      $v0,$zero,$L38
        lw       $v0,152($fp)
        slt      $v0,$v0,123
        bne      $v0,$zero,$L35
$L38:
        lw       $v1,152($fp)
        li       $v0,95                # 0x5f
        beq      $v1,$v0,$L35
        lw       $v1,152($fp)
        li       $v0,45                # 0x2d
        beq      $v1,$v0,$L35
        addu     $v0,$fp,40
        move     $a0,$v0
        move     $a1,$zero
        li       $a2,100               # 0x64
        la       $t9,memset
        jal      $ra,$t9
        sw       $zero,148($fp)
$L43:
        lw       $v1,144($fp)
        lw       $v0,28($fp)
        subu     $v1,$v1,$v0
        lw       $v0,148($fp)
        slt      $v0,$v0,$v1
        bne      $v0,$zero,$L46
        b        $L44
$L46:
        addu     $v1,$fp,40
        lw       $v0,148($fp)
        addu     $a0,$v1,$v0
        lw       $v1,28($fp)
        lw       $v0,148($fp)
        addu     $v1,$v1,$v0
        lw       $v0,176($fp)
        addu     $v0,$v1,$v0
        lbu      $v0,0($v0)
        sb       $v0,0($a0)
        lw       $v0,148($fp)
        addu     $v0,$v0,1
        sw       $v0,148($fp)
        b        $L43
$L44:
        addu     $v0,$fp,40
        move     $a0,$v0
        la       $t9,es_capicua

```



```

jal      $ra,$t9
beq      $v0,$zero,$L47
sw       $zero,148($fp)
$L48:
lw       $v1,144($fp)
lw       $v0,28($fp)
subu     $v1,$v1,$v0
lw       $v0,148($fp)
slt      $v0,$v0,$v1
bne      $v0,$zero,$L51
b        $L49
$L51:
lw       $v1,180($fp)
lw       $v0,32($fp)
addu     $a0,$v1,$v0
addu     $v1,$fp,40
lw       $v0,148($fp)
addu     $v0,$v1,$v0
lbu      $v0,0($v0)
sb       $v0,0($a0)
lw       $v0,32($fp)
addu     $v0,$v0,1
sw       $v0,32($fp)
lw       $v0,148($fp)
addu     $v0,$v0,1
sw       $v0,148($fp)
b        $L48
$L49:
lw       $v1,180($fp)
lw       $v0,32($fp)
addu     $v1,$v1,$v0
li       $v0,10
sb       $v0,0($v1)
lw       $v0,32($fp)
addu     $v0,$v0,1
sw       $v0,32($fp)
lw       $v0,36($fp)
addu     $v0,$v0,1
sw       $v0,36($fp)
$L47:
lw       $v0,144($fp)
addu     $v0,$v0,1
sw       $v0,28($fp)
$L35:
lw       $v0,144($fp)
addu     $v0,$v0,1

```

0xa

```

sw      $v0,144($fp)
b       $L33
$L34:
lw      $v0,36($fp)
move    $sp,$fp
lw      $ra,168($sp)
lw      $fp,164($sp)
addu    $sp,$sp,176
j       $ra
.end     palabras_en_linea
.size    palabras_en_linea, .-palabras_en_linea
.rdata
.align  2
$LC2:
.ascii  "help\000"
.align  2
$LC3:
.ascii  "version\000"
.align  2
$LC4:
.ascii  "input\000"
.align  2
$LC5:
.ascii  "output\000"
.data
.align  2
.type   long_options.0, @object
.size   long_options.0, 80
long_options.0:
.word   $LC2
.word   0
.word   0
.word   104
.word   $LC3
.word   0
.word   0
.word   118
.word   $LC4
.word   1
.word   0
.word   105
.word   $LC5
.word   1
.word   0
.word   111
.word   0

```

```

        .word    0
        .word    0
        .word    0
        .rdata
        .align   2
$LC6:
        .ascii   "hvc:i:o:\000"
        .align   2
$LC7:
        .ascii   "option %s\000"
        .align   2
$LC8:
        .ascii   " with arg %s\000"
        .align   2
$LC9:
        .ascii   "\n\000"
        .align   2
$LC10:
        .ascii   "non-option ARGV-elements: \000"
        .align   2
$LC11:
        .ascii   "%s \000"
        .align   2
$LC12:
        .ascii   "r\000"
        .align   2
$LC13:
        .ascii   "Ingrese la oracion a evaluar:\n\000"
        .align   2
$LC14:
        .ascii   "w\000"
        .align   2
$LC15:
        .ascii   "Palabras capicua:\n"
        .ascii   "%s\000"
        .text
        .align   2
        .globl   main
        .ent     main
main:
        .frame   $fp,4176,$ra                # vars= 4128, regs= 3/0, args= 24, ex
        .mask    0xd0000000,-8
        .fmask   0x00000000,0
        .set     noreorder
        .cpld    $t9
        .set     reorder

```

```

subu    $sp,$sp,4176
.cprestore 24
sw      $ra,4168($sp)
sw      $fp,4164($sp)
sw      $gp,4160($sp)
move    $fp,$sp
sw      $a0,4176($fp)
sw      $a1,4180($fp)
sw      $zero,32($fp)
sw      $zero,36($fp)

$L53:
sw      $zero,44($fp)
addu    $v0,$fp,44
sw      $v0,16($sp)
lw      $a0,4176($fp)
lw      $a1,4180($fp)
la      $a2,$LC6
la      $a3,long_options.0
la      $t9,getopt_long
jal     $ra,$t9
sw      $v0,40($fp)
lw      $v1,40($fp)
li      $v0,-1
bne     $v1,$v0,$L56
b       $L54

$L56:
lw      $v0,40($fp)
sw      $v0,4156($fp)
li      $v0,105
lw      $v1,4156($fp)
beq     $v1,$v0,$L63
lw      $v1,4156($fp)
slt     $v0,$v1,106
beq     $v0,$zero,$L68
li      $v0,104
lw      $v1,4156($fp)
beq     $v1,$v0,$L61
lw      $v0,4156($fp)
beq     $v0,$zero,$L58
li      $v0,63
lw      $v1,4156($fp)
beq     $v1,$v0,$L53
b       $L66

$L68:
li      $v0,111
lw      $v1,4156($fp)

```

```

        beq    $v1,$v0,$L64
        li     $v0,118                                # 0x76
        lw     $v1,4156($fp)
        beq    $v1,$v0,$L62
        b      $L66
$L58:
        lw     $v0,44($fp)
        sll    $v1,$v0,4
        la     $v0,long_options.0+8
        addu   $v0,$v1,$v0
        lw     $v0,0($v0)
        beq    $v0,$zero,$L59
        b      $L53
$L59:
        lw     $v0,44($fp)
        sll    $v1,$v0,4
        la     $v0,long_options.0
        addu   $v0,$v1,$v0
        la     $a0,$LC7
        lw     $a1,0($v0)
        la     $t9,printf
        jal    $ra,$t9
        lw     $v0,optarg
        beq    $v0,$zero,$L60
        la     $a0,$LC8
        lw     $a1,optarg
        la     $t9,printf
        jal    $ra,$t9
$L60:
        la     $a0,$LC9
        la     $t9,printf
        jal    $ra,$t9
        b      $L53
$L61:
        la     $t9,print_help
        jal    $ra,$t9
        move   $a0,$zero
        la     $t9,exit
        jal    $ra,$t9
$L62:
        la     $t9,print_version
        jal    $ra,$t9
        move   $a0,$zero
        la     $t9,exit
        jal    $ra,$t9
$L63:

```

```

        lw      $v0,optarg
        sw      $v0,36($fp)
        b       $L53
$L64:
        lw      $v0,optarg
        sw      $v0,32($fp)
        b       $L53
$L66:
        la      $t9,abort
        jal     $ra,$t9
$L54:
        lw      $v0,optind
        lw      $v1,4176($fp)
        slt     $v0,$v0,$v1
        beq     $v0,$zero,$L69
        la      $a0,$LC10
        la      $t9,printf
        jal     $ra,$t9
$L70:
        lw      $v0,optind
        lw      $v1,4176($fp)
        slt     $v0,$v0,$v1
        bne     $v0,$zero,$L72
        b       $L71
$L72:
        la      $a1,optind
        lw      $v1,0($a1)
        move    $v0,$v1
        sll     $a0,$v0,2
        lw      $v0,4180($fp)
        addu    $v0,$a0,$v0
        addu    $v1,$v1,1
        sw      $v1,0($a1)
        la      $a0,$LC11
        lw      $a1,0($v0)
        la      $t9,printf
        jal     $ra,$t9
        b       $L70
$L71:
        li      $a0,10
        la      $a1,--sF+88
        la      $t9,--sputc
        jal     $ra,$t9
$L69:
        addu    $v0,$fp,1072
        move    $a0,$v0

```

0xa

```

move    $a1,$zero
li      $a2,1024                                # 0x400
la      $t9,memset
jal     $ra,$t9
lw      $a0,36($fp)
la      $a1,$LC12
la      $t9,fopen
jal     $ra,$t9
sw      $v0,2096($fp)
lw      $v0,2096($fp)
bne     $v0,$zero,$L75
la      $a0,$LC13
la      $t9,printf
jal     $ra,$t9
addu    $v0,$fp,48
move    $a0,$v0
li      $a1,1024                                # 0x400
la      $a2,--sF
la      $t9,fgets
jal     $ra,$t9
addu    $v0,$fp,48
addu    $v1,$fp,1072
move    $a0,$v0
move    $a1,$v1
la      $t9,palabras_en_linea
jal     $ra,$t9
b       $L74
$L75:
lw      $v0,2096($fp)
lhu     $v0,12($v0)
srl     $v0,$v0,5
andi    $v0,$v0,0x1
beq     $v0,$zero,$L77
b       $L76
$L77:
addu    $v0,$fp,2104
move    $a0,$v0
move    $a1,$zero
li      $a2,1024                                # 0x400
la      $t9,memset
jal     $ra,$t9
addu    $v0,$fp,2104
move    $a0,$v0
li      $a1,1024                                # 0x400
lw      $a2,2096($fp)
la      $t9,fgets

```

```

jal    $ra,$t9
lb     $v0,2104($fp)
bne    $v0,$zero,$L78
b      $L75
$L78:
addu   $v0,$fp,3128
move   $a0,$v0
move   $a1,$zero
li     $a2,1024                # 0x400
la     $t9,memset
jal    $ra,$t9
addu   $v0,$fp,2104
addu   $v1,$fp,3128
move   $a0,$v0
move   $a1,$v1
la     $t9,palabras_en_linea
jal    $ra,$t9
addu   $v0,$fp,1072
addu   $v1,$fp,3128
move   $a0,$v0
move   $a1,$v1
la     $t9,strcmp
jal    $ra,$t9
b      $L75
$L76:
lw     $a0,2096($fp)
la     $t9,fclose
jal    $ra,$t9
$L74:
lw     $a0,32($fp)
la     $a1,$LC14
la     $t9,fopen
jal    $ra,$t9
sw     $v0,4152($fp)
lw     $v0,4152($fp)
bne    $v0,$zero,$L79
addu   $v0,$fp,1072
la     $a0,$LC15
move   $a1,$v0
la     $t9,printf
jal    $ra,$t9
b      $L80
$L79:
addu   $v0,$fp,1072
move   $a0,$v0
lw     $a1,4152($fp)

```



```

        la      $t9, fputs
        jal     $ra, $t9
        lw      $a0, 4152($fp)
        la      $t9, fclose
        jal     $ra, $t9
$L80:
        move    $v0, $zero
        move    $sp, $fp
        lw      $ra, 4168($sp)
        lw      $fp, 4164($sp)
        addu    $sp, $sp, 4176
        j       $ra
        .end    main
        .size   main, .-main
        .align  2
        .ent    __sputc
__sputc:
        .frame   $fp, 48, $ra
        .mask    0xd0000000, -8
        .fmask   0x00000000, 0
        .set     noreorder
        .cload   $t9
        .set     reorder
        subu     $sp, $sp, 48
        .cprestore 16
        sw       $ra, 40($sp)
        sw       $fp, 36($sp)
        sw       $gp, 32($sp)
        move     $fp, $sp
        sw       $a0, 48($fp)
        sw       $a1, 52($fp)
        lw       $v1, 52($fp)
        lw       $v0, 52($fp)
        lw       $v0, 8($v0)
        addu     $v0, $v0, -1
        sw       $v0, 8($v1)
        bgez     $v0, $L3
        lw       $v0, 52($fp)
        lw       $v1, 52($fp)
        lw       $a0, 8($v0)
        lw       $v0, 24($v1)
        slt      $v0, $a0, $v0
        bne      $v0, $zero, $L2
        lb       $v1, 48($fp)
        li       $v0, 10
        bne      $v1, $v0, $L3

```

vars= 8, regs= 3/0, args= 16, extra=

0xa

```

    b        $L2
$L3:
    lw        $a1,52($fp)
    lw        $v1,0($a1)
    lbu       $a0,48($fp)
    move      $v0,$v1
    sb        $a0,0($v0)
    andi      $v0,$a0,0x00ff
    addu      $v1,$v1,1
    sw        $v1,0($a1)
    sw        $v0,24($fp)
    b        $L1
$L2:
    lw        $a0,48($fp)
    lw        $a1,52($fp)
    la        $t9,--swbuf
    jal       $ra,$t9
    sw        $v0,24($fp)
$L1:
    lw        $v0,24($fp)
    move      $sp,$fp
    lw        $ra,40($sp)
    lw        $fp,36($sp)
    addu      $sp,$sp,48
    j         $ra
    .end      __putc
    .size     __putc,.-__putc
    .ident    "GCC: (GNU) 3.3.3 (NetBSD nb3 20040520)"

```

4. Corridas de prueba

4.1. Primer corrida

Se corrió el programa con la siguiente entrada:

```
Somos los primeros en completar el TP 0.  
Ojo que La fecha de entrega del TP0 es el martes 12 de septiembre.
```

Se obtuvo la siguiente salida:

```
Somos  
Ojo
```

4.2. Segunda corrida

Se corrió el programa con la siguiente entrada:

```
MeNEm neUquEn 1a2d323d2a1 adke  
pepe$nene/larral=dom-mod?a23_32a
```

Se obtuvo la siguiente salida:

```
MeNEm  
neUquEn  
1a2d323d2a1  
larral  
dom-mod  
a23_32a
```

4.3. Tercera corrida

Se corrió el programa con la siguiente entrada:

```
aD-2eT_R_Te2-Da%4004?CheVr  
peep23*** avion{daad}  
neUqUeN&NarNran
```

Se obtuvo la siguiente salida:

```
aD-2eT_R_Te2-Da  
4004  
daad  
neUqUeN  
NarNran
```