

Trabajo práctico 0: Infraestructura básica

Roberto Herman, *Padrón Nro. 84.803*
`berta1108@gmail.com`

Matías Waisgold, *Padrón Nro. 88.464*
`mwaisgold@gmail.com`

Nicolás Suarez, *Padrón Nro. 83.752*
`nicolas.suarez@gmail.com`

Federico Rodriguez, *Padrón Nro. 88.310*
`fedel726@gmail.com`

Grupo Nro. 0 - 2do. Cuatrimestre de 2009
66.20 Organización de Computadoras
Facultad de Ingeniería, Universidad de Buenos Aires

1. Introducción

El presente trabajo práctico consiste en una introducción a la utilización de la infraestructura utilizada para el desarrollo de los trabajos prácticos de esta materia. La misma será llevada a cabo a través del desarrollo de una versión reducida del comando **tail**[1] de UNIX. El comando será codificado en lenguaje C, y se compilará para funcionar en el emulador de **MIPS32 NetBsd**.

2. Diseño e implementación

2.1. Módulo principal (*main*)

El módulo principal *tp0.c* contiene la validación de parámetros de entrada, así como el parseo de los mismos.

2.2. Parser de líneas (*readLines*)

Esta función se encarga de procesar un archivo de texto especificado para luego mostrar las últimas **n** líneas del mismo.

2.3. Parser de bytes (*readBytes*)

Esta función se encarga de procesar un archivo de texto especificado para luego mostrar los últimos **n** bytes del mismo.

2.4. Lógica de parseo

Para la función de parseo de líneas, la lógica consiste en tener dos punteros al archivo en cuestión. Inicialmente el primer puntero avanza, mientras que el otro espera a ser sobrepasado por éste, una vez ocurrido esto, se incrementa la posición de este segundo puntero. Finalmente, cuando el puntero que está avanzando llega al final del archivo, se lee desde la posición de donde estaba el otro puntero, que es el que tiene la cantidad exacta de líneas a procesar. Para la función de parseo de bytes, el procedimiento es más sencillo. Según los parámetros especificados, se utiliza un sólo puntero para recorrer el archivo y calcular la cantidad de bytes pedida.

3. Instrucciones de uso

El uso de la aplicación es idéntico al propuesto en el enunciado del trabajo práctico [2].

4. Comandos

En esta sección se describen los comandos necesarios para la compilación de la aplicación.

4.1. Generando el programa ejecutable

```
gcc -Wall -o tp0 tp0.c
```

4.2. Obteniendo el código MIPS generado por el compilador

```
gcc -Wall -O0 -S -mrnames tp0.c
```

5. Corridas de prueba

- Prueba 1: Imprimimos la opcion h (ayuda) y la version (V).

```
root@:~# ./tp0 -h
Uso: ./tp0 opciones [ argumentos ...] NOMBRE_ARCHIVO
    -h  --help                Muestra esta ayuda y finaliza
    -n  --lines lineas        Cantidad de lineas a leer
    -c  --bytes                Lee la cantidad de bytes en vez de las lineas

root@:~# ./tp0 -V
Version [66.20] Organizacion de Computadoras
Segundo Cuatrimestre 2009
```

- Prueba 2: Imprimimos las 3 ultimas lineas de google.txt.

```
root@:~# ./tp0 -n3 google.txt
para que puedan ser vistas a través del software Google Earth, cuyo
objetivo es permitir a los usuarios del mencionado software aprender
más sobre una zona específica del mapa, observando las fotografías
que otros usuarios han tomado ahí.
```

Comando tail:

```
root@:~# ./tp0 -n3 google.txt
para que puedan ser vistas a través del software Google Earth, cuyo
objetivo es permitir a los usuarios del mencionado software aprender
más sobre una zona específica del mapa, observando las fotografías
que otros usuarios han tomado ahí.
```

- Prueba 3: Imprimimos google.txt sin parametros. Imprime últimas 10 lineas por default.

Imprimimos google.txt sin parametros. Por default imprime las ultimas 10 lineas.

```
root@:~# ./tp0 google.txt
En abril de 2007, Google compró DoubleClick,[13] una empresa especializada
en publicidad enInternet, por 3.100 millones de dólares.
Este mismo mes, Google se convirtió en la marca más valiosa del mundo,
alcanzando la suma de 66.000 millones de dólares, superando a emblemáticas
empresas como Microsoft, General Electric y Coca-Cola.
```

En julio de 2007, Google compró Panoramio, un sitio web dedicado a exhibir las fotografías que los propios usuarios crean y geoposicionan, siendo algunas de ellas subidas al sitio para que puedan ser vistas a través del software Google Earth, cuyo objetivo es permitir a los usuarios del mencionado software aprender más sobre una zona específica del mapa, observando las fotografías que otros usuarios han tomado ahí.

Comando tail:

```
fed@Fede-PC:~/Escritorio$ tail google.txt
```

En abril de 2007, Google compró DoubleClick,[13] una empresa especializada en publicidad enInternet, por 3.100 millones de dólares.

Este mismo mes, Google se convirtió en la marca más valiosa del mundo, alcanzando la suma de 66.000 millones de dólares, superando a emblemáticas empresas como Microsoft, General Electric y Coca-Cola.

En julio de 2007, Google compró Panoramio, un sitio web dedicado a exhibir las fotografías que los propios usuarios crean y geoposicionan, siendo algunas de ellas subidas al sitio para que puedan ser vistas a través del software Google Earth, cuyo objetivo es permitir a los usuarios del mencionado software aprender más sobre una zona específica del mapa, observando las fotografías que otros usuarios han tomado ahí.

- Prueba 4: Imprimimos la ultima linea de google.txt, Beethoven.txt y codigo_barras.txt

```
root@:~# ./tp0 -n1 google.txt Beethoven.txt codigo_de_barras.txt
```

```
==> google.txt <==
```

observando las fotografías que otros usuarios han tomado ahí.

```
==> Beethoven.txt <==
```

de los hijos de María Magdalena Keverich.

```
==> codigo_de_barras.txt <==
```

X, 2X, 3X, y 4X.

Comando tail:

```
fed@Fede-PC:~/Escritorio$ tail -n1 google.txt Beethoven.txt codigo_de_barras.txt
```

```
==> google.txt <==
```

observando las fotografías que otros usuarios han tomado ahí.

```
==> Beethoven.txt <==
```

de los hijos de María Magdalena Keverich.

```
==> codigo_de_barras.txt <==
```

X, 2X, 3X, y 4X.

- Prueba 5: Imprimimos los ultimos 30 bytes de google.txt.

```
root@:~# ./tp0 -c50 google.txt
fotografías que otros usuarios han tomado ahí.
```

```
Comando tail:
fed@Fede-PC:~/Escritorio$ tail -c50 google.txt
fotografías que otros usuarios han tomado ahí.
```

- Prueba 6: Imprimimos los ultimos 100 bytes de Beethoven.txt y google.txt.

```
root@:~# ./tp0 -c100 Beethoven.txt google.txt
```

```
==> Beethoven.txt <==
ienen claras las fechas de nacimiento exactas de ninguno
de los hijos de María Magdalena Keverich.
```

```
==> google.txt <==
obre una zona específica del mapa,
observando las fotografías que otros usuarios han tomado ahí.
```

```
Comando tail:
fed@Fede-PC:~/Escritorio$ tail -c100 Beethoven.txt google.txt
==> Beethoven.txt <==
ienen claras las fechas de nacimiento exactas de ninguno
de los hijos de María Magdalena Keverich.
```

```
==> google.txt <==
obre una zona específica del mapa,
observando las fotografías que otros usuarios han tomado ahí.
```

Se adjunta en el entregable de ejecutable y documentación, un archivo de texto de prueba.

6. Conclusiones

Este trabajo práctico consistió en un primer enfoque en la infraestructura que será utilizada para desarrollar futuras aplicaciones en la cursada. Si bien la aplicación a programar no ofrecía mayor complejidad, el desarrollo del trabajo práctico consumió más tiempo de lo esperado, más que nada debido a que parte del tiempo de trabajo fue ocupado por la configuración inicial del entorno Linux para trabajar con el emulador de MIPS32 NetBSD (instalación del emulador, configuración del tunel, comandos propios del entorno, etc). Tareas tales como la generación del código assembly MIPS a través del código fuente en lenguaje C, y la ejecución del programa directamente en el emulador, nos permitió familiarizarnos con este entorno, desconocido para la mayoría de los integrantes de este grupo.

Referencias

- [1] tail: <http://www.oreillynet.com/linux/cmd/cmd.csp?path=t/tail>

A. Apéndice: Código Fuente

A.1. Lenguaje C

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

#define FIRST_OPTION 0
#define SECOND_OPTION 1

/*Muestra las ultimas de un archivo
 * Parametros:
 * path: ruta del archivo a leer
 * lines: cantidad de lineas a leer
 * option: si option=0 lee las ultimas lineas, si option=1 lee desde
 *          la linea igual a lines*/
void readLines(char *path, int lines, int option)
{
    FILE *fd1, *fd2;
    int countLines=0;
    char buffer, lastChar;

    fd1 = fopen(path, "r");
    if(fd1 == NULL)
    {
        printf("No se pudo abrir el archivo: %s", path);
        return;
    }

    if( option == SECOND_OPTION )
    {
        countLines = 1;
        buffer = (char) fgetc(fd1);
        while(!feof(fd1))
        {
            if(countLines == lines)
                printf("%c", buffer);
            else if(buffer == '\n')
                countLines++;
            buffer = (char) fgetc(fd1);
        }
        fclose(fd1);
        return;
    }

    fd2 = fopen(path, "r");
    if(fd1 == NULL)
```

```

    {
        printf("No se pudo abrir el archivo: %s", path);
        fclose(fd1);
        return;
    }

    buffer = (char) fgetc(fd1);
    while(!feof(fd1))
    {
        if(buffer == '\n')
        {
            if(countLines < lines)
                countLines++;
            else
                while((char)fgetc(fd2) != '\n')
                {
                }
        }
        lastChar = buffer;
        buffer = (char) fgetc(fd1);
    }
    if(countLines == lines && lastChar != '\n')
        while((char)fgetc(fd2) != '\n')
        {
        }
    buffer = (char) fgetc(fd2);
    while(!feof(fd2))
    {
        printf("%c", buffer);
        buffer = (char) fgetc(fd2);
    }
    fclose(fd1);
    fclose(fd2);
}

/*Muestra los bytes de un archivo
 * Parametros:
 * path: ruta del archivo a leer
 * bytes: cantidad de bytes a leer
 * option: si option=0 lee los ultimos bytes, si option=1 lee desde
 * el byte igual a bytes*/
void readBytes(char *path, int bytes, int option)
{
    FILE *fd;
    char buffer;
    int sizeFile = 0;
    fd = fopen(path, "rb");
    if(fd == NULL)
    {

```

```

        printf("No se pudo abrir el archivo: %s", path);
        return;
    }
    fseek(fd, 0, SEEK_END);
    sizeFile = ftell(fd);
    if(option == SECOND_OPTION)
    {
        /* tiene q empezar a leer en una posicion mayor al tamaño del archivo
        * entonces no lee nada*/
        if (bytes >= sizeFile)
            return;
        else
            fseek(fd, bytes, SEEK_SET);
    }
    else
    {
        if (bytes > sizeFile)
            return;
        fseek(fd, sizeFile - bytes, SEEK_SET);
    }
    buffer = (char) fgetc(fd);
    while(!feof(fd))
    {
        printf("%c", buffer);
        buffer = (char) fgetc(fd);
    }
    fclose(fd);
}

const char* nombre_programa;

void imprime_uso (){
    printf("Uso: %s opciones [ argumentos ...] NOMBRE_ARCHIVO\n", nombre_programa);
    printf("    -h --help                Muestra esta ayuda y finaliza\n"
           "    -n --lines lines          Cantidad de lineas a leer\n"
           "    -c --bytes                Lee la cantidad de bytes en vez de las lineas\n");
}

void imprime_version(){
    printf("Version [66.20] Organizacion de Computadoras\n"
           "Segundo Cuatrimestre 2009");
}

/*define la cantidad de lineas a leer, y la opcion de lectura*/
void definirLineas(int optind, int argc, char* argv[],
                   int cantLineas, int showBytes, int readingOption){

    char* file = NULL;

```



```

/*Las siguiente variables se usan para leer la stdin*/
char* tempFile = "temp_tail.txt";
char c = '\0';
FILE *fd;

/* Guarda la posicion del siguiente archivo a leer*/
int siguiente_archivo = optind;

while(siguiente_archivo <= argc)
{
    if (siguiente_archivo < argc){
        file = argv[siguiente_archivo];
    }
    else /*Si no se paso un archivo por parametro lee de la stdin*/
    {
        if(optind < argc)
            break;
        file = tempFile;
        fd = fopen(file, "wb");
        c = fgetc(stdin);
        while(c != EOF)
        {
            fputc(c, fd);
            c = fgetc(stdin);
        }
        fclose(fd);
    }

    if (file == NULL){
        imprime_uso();
        exit(1);
    }

    /*Si hay mas de un archivo a leer, muestra el nombre del archivo*/
    if(optind + 1 < argc)
    {
        printf("\n==> %s <==\n", file);
    }

    if(showBytes)
        readBytes(file, cantLineas, readingOption);
    else
        readLines(file, cantLineas, readingOption);
    siguiente_archivo++;
}
if(optind == argc)
    remove(tempFile);
}

```

```

int main(int argc, char* argv[])
{
    int siguiente_opcion = 0;

    /* Una cadena que lista las opciones cortas validas */
    const char* const op_cortas = "hc:n:V" ;

    /* Una estructura de varios arrays describiendo los valores largos */
    const struct option op_largas[] =
    {
        { "help",          0,  NULL,  'h'},
        { "bytes",         1,  NULL,  'c'},
        { "lines",         1,  NULL,  'n'},
        { "version",       0,  NULL,  'V'},
        { NULL,            0,  NULL,  0  }
    };

    int showBytes = 0; //Por defecto muestra las ultimas lineas
    int cantLineas = 10; //Le pongo por default 10 lineas
    int readingOption = FIRST_OPTION;

    /* Guardar el nombre del programa para incluirlo a la salida */
    nombre_programa = argv[0];

    while (siguiente_opcion != -1)
    {
        /* Llamamos a la funcion getopt */
        siguiente_opcion = getopt_long (argc, argv, op_cortas, op_largas, NULL);

        switch (siguiente_opcion)
        {
            case 'h' : /* -h o --help */
                imprime_uso();
                exit(EXIT_SUCCESS);

            case 'V' : /* -V o --version*/
                imprime_version();
                exit(EXIT_SUCCESS);

            case 'c' :
                showBytes = 1;
                if(optarg[0]=='+')
                {
                    cantLineas = atoi(optarg+1);
                    readingOption = SECOND_OPTION;
                }
                else
                    cantLineas = atoi(optarg);
                break;
        }
    }
}

```

```

        case 'n' :
            showBytes = 0;
            if(optarg[0]=='+')
            {
                cantLineas = atoi(optarg+1);
                readingOption = SECOND_OPTION;
            }
            else
                cantLineas = atoi(optarg);
            break;

        case '?' : /* opcion no valida */
            imprime_uso(); /* codigo de salida 1 */
            exit(1);

        case -1 : /* No hay mas opciones */
            break;

        default : /* Algo mas? No esperado. Abortamos */
            abort();
    }
}
definirLineas(optind, argc, argv,
              cantLineas, showBytes, readingOption);
return 0;
}

```

A.2. Lenguaje MIPS

```

.file 1 "tp0.c"
.section .mdebug.abi32
.previous
.abicalls
.rdata
.align 2
$LC0:
.ascii "r\000"
.align 2
$LC1:
.ascii "No se pudo abrir el archivo: %s\000"
.align 2
$LC2:
.ascii "%c\000"
.text
.align 2
.globl readLines
.ent readLines
readLines:

```

```

.frame $fp,56,$ra # vars= 16, regs= 3/0, args= 16, extra= 8
.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)
sw $a1,60($fp)
sw $a2,64($fp)
sw $zero,32($fp)
lw $a0,56($fp)
la $a1,$LC0
la $t9,fopen
jal $ra,$t9
sw $v0,24($fp)
lw $v0,24($fp)
bne $v0,$zero,$L18
la $a0,$LC1
lw $a1,56($fp)
la $t9,printf
jal $ra,$t9
b $L17
$L18:
lw $v1,64($fp)
li $v0,1 # 0x1
bne $v1,$v0,$L19
li $v0,1 # 0x1
sw $v0,32($fp)
lw $a0,24($fp)
la $t9,fgetc
jal $ra,$t9
sb $v0,36($fp)
$L20:
lw $v0,24($fp)
lhu $v0,12($v0)
srl $v0,$v0,5
andi $v0,$v0,0x1
beq $v0,$zero,$L22
b $L21
$L22:
lw $v1,32($fp)
lw $v0,60($fp)
bne $v1,$v0,$L23
lb $v0,36($fp)

```

```

la $a0,$LC2
move $a1,$v0
la $t9,printf
jal $ra,$t9
b $L24
$L23:
lb $v1,36($fp)
li $v0,10 # 0xa
bne $v1,$v0,$L24
lw $v0,32($fp)
addu $v0,$v0,1
sw $v0,32($fp)
$L24:
lw $a0,24($fp)
la $t9,fgetc
jal $ra,$t9
sb $v0,36($fp)
b $L20
$L21:
lw $a0,24($fp)
la $t9,fclose
jal $ra,$t9
b $L17
$L19:
lw $a0,56($fp)
la $a1,$LC0
la $t9,fopen
jal $ra,$t9
sw $v0,28($fp)
lw $v0,24($fp)
bne $v0,$zero,$L26
la $a0,$LC1
lw $a1,56($fp)
la $t9,printf
jal $ra,$t9
lw $a0,24($fp)
la $t9,fclose
jal $ra,$t9
b $L17
$L26:
lw $a0,24($fp)
la $t9,fgetc
jal $ra,$t9
sb $v0,36($fp)
$L27:
lw $v0,24($fp)
lhu $v0,12($v0)
srl $v0,$v0,5
andi $v0,$v0,0x1
beq $v0,$zero,$L29

```

```

b $L28
$L29:
lb $v1,36($fp)
li $v0,10 # 0xa
bne $v1,$v0,$L30
lw $v0,32($fp)
lw $v1,60($fp)
slt $v0,$v0,$v1
beq $v0,$zero,$L31
lw $v0,32($fp)
addu $v0,$v0,1
sw $v0,32($fp)
b $L30
$L31:
.set noreorder
nop
.set reorder
$L33:
lw $a0,28($fp)
la $t9,fgetc
jal $ra,$t9
sll $v0,$v0,24
sra $v1,$v0,24
li $v0,10 # 0xa
bne $v1,$v0,$L33
$L30:
lw $a0,24($fp)
la $t9,fgetc
jal $ra,$t9
sb $v0,36($fp)
b $L27
$L28:
lw $v1,32($fp)
lw $v0,60($fp)
bne $v1,$v0,$L36
$L37:
lw $a0,28($fp)
la $t9,fgetc
jal $ra,$t9
sll $v0,$v0,24
sra $v1,$v0,24
li $v0,10 # 0xa
bne $v1,$v0,$L37
$L36:
lw $a0,28($fp)
la $t9,fgetc
jal $ra,$t9
sb $v0,36($fp)
$L40:
lw $v0,28($fp)

```

```

    lhu $v0,12($v0)
    srl $v0,$v0,5
    andi $v0,$v0,0x1
    beq $v0,$zero,$L42
    b $L41
$L42:
    lb $v0,36($fp)
    la $a0,$LC2
    move $a1,$v0
    la $t9,printf
    jal $ra,$t9
    lw $a0,28($fp)
    la $t9,fgetc
    jal $ra,$t9
    sb $v0,36($fp)
    b $L40
$L41:
    lw $a0,24($fp)
    la $t9,fclose
    jal $ra,$t9
    lw $a0,28($fp)
    la $t9,fclose
    jal $ra,$t9
$L17:
    move $sp,$fp
    lw $ra,48($sp)
    lw $fp,44($sp)
    addu $sp,$sp,56
    j $ra
.end readLines
.size readLines, .-readLines
.rdata
.align 2
$LC3:
.ascii "rb\000"
.text
.align 2
.globl readBytes
.ent readBytes
readBytes:
.frame $fp,56,$ra # vars= 16, regs= 3/0, args= 16, extra= 8
.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)
sw $a1,60($fp)
sw $a2,64($fp)
sw $zero,32($fp)
lw $a0,56($fp)
la $a1,$LC3
la $t9,fopen
jal $ra,$t9
sw $v0,24($fp)
lw $v0,24($fp)
bne $v0,$zero,$L44
la $a0,$LC1
lw $a1,56($fp)
la $t9,printf
jal $ra,$t9
b $L43
$L44:
lw $a0,24($fp)
move $a1,$zero
li $a2,2 # 0x2
la $t9,fseek
jal $ra,$t9
lw $a0,24($fp)
la $t9,ftell
jal $ra,$t9
sw $v0,32($fp)
lw $v1,64($fp)
li $v0,1 # 0x1
bne $v1,$v0,$L45
lw $v0,60($fp)
lw $v1,32($fp)
slt $v0,$v0,$v1
bne $v0,$zero,$L46
b $L43
$L46:
lw $a0,24($fp)
lw $a1,60($fp)
move $a2,$zero
la $t9,fseek
jal $ra,$t9
b $L48
$L45:
lw $v0,60($fp)
lw $v1,32($fp)
slt $v0,$v1,$v0
beq $v0,$zero,$L49
b $L43
$L49:

```



```

lw $v0,32($fp)
lw $v1,60($fp)
subu $v0,$v0,$v1
lw $a0,24($fp)
move $a1,$v0
move $a2,$zero
la $t9,fseek
jal $ra,$t9
$L48:
lw $a0,24($fp)
la $t9,fgetc
jal $ra,$t9
sb $v0,28($fp)
$L50:
lw $v0,24($fp)
lhu $v0,12($v0)
srl $v0,$v0,5
andi $v0,$v0,0x1
beq $v0,$zero,$L52
b $L51
$L52:
lb $v0,28($fp)
la $a0,$LC2
move $a1,$v0
la $t9,printf
jal $ra,$t9
lw $a0,24($fp)
la $t9,fgetc
jal $ra,$t9
sb $v0,28($fp)
b $L50
$L51:
lw $a0,24($fp)
la $t9,fclose
jal $ra,$t9
$L43:
move $sp,$fp
lw $ra,48($sp)
lw $fp,44($sp)
addu $sp,$sp,56
j $ra
.end readBytes
.size readBytes, .-readBytes
.rdata
.align 2
$LC4:
.ascii "Uso: %s opciones [ argumentos ...] NOMBRE_ARCHIVO\n\000"
.align 2
$LC5:
.ascii "      -h  --help                Muestra esta ayuda y fin"

```

```

.ascii "aliza\n"
.ascii "    -n  --lines lineas          Cantidad de lineas a lee"
.ascii "r\n"
.ascii "    -c  --bytes                Lee la cantidad de bytes"
.ascii " en vez de las lineas\n\000"
.text
.align 2
.globl imprime_uso
.ent imprime_uso
imprime_uso:
.frame $fp,40,$ra # vars= 0, regs= 3/0, args= 16, extra= 8
.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 $a0,$LC4
lw $a1,nombre_programa
la $t9,printf
jal $ra,$t9
la $a0,$LC5
la $t9,printf
jal $ra,$t9
move $sp,$fp
lw $ra,32($sp)
lw $fp,28($sp)
addu $sp,$sp,40
j $ra
.end imprime_uso
.size imprime_uso, .-imprime_uso
.rdata
.align 2
$LC6:
.ascii "Version [66.20] Organizacion de Computadoras\n"
.ascii "Segundo Cuatrimestre 2009\000"
.text
.align 2
.globl imprime_version
.ent imprime_version
imprime_version:
.frame $fp,40,$ra # vars= 0, regs= 3/0, args= 16, extra= 8
.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 $a0,$LC6
la $t9,printf
jal $ra,$t9
move $sp,$fp
lw $ra,32($sp)
lw $fp,28($sp)
addu $sp,$sp,40
j $ra
.end imprime_version
.size imprime_version, .-imprime_version
.rdata
.align 2
$LC7:
.ascii "temp_tail.txt\000"
.align 2
$LC8:
.ascii "wb\000"
.align 2
$LC9:
.ascii "=="> %s <==\n\000"
.align 2
$LC10:
.ascii "\n\n\000"
.text
.align 2
.globl definirLineas
.ent definirLineas
definirLineas:
.frame $fp,64,$ra # vars= 24, regs= 3/0, args= 16, extra= 8
.mask 0xd0000000,-8
.fmask 0x00000000,0
.set noreorder
.cpload $t9
.set reorder
subu $sp,$sp,64
.cprestore 16
sw $ra,56($sp)
sw $fp,52($sp)
sw $gp,48($sp)
move $fp,$sp
sw $a0,64($fp)
sw $a1,68($fp)

```

```

sw $a2,72($fp)
sw $a3,76($fp)
sw $zero,24($fp)
la $v0,$LC7
sw $v0,28($fp)
sb $zero,32($fp)
lw $v0,64($fp)
sw $v0,40($fp)
$L56:
lw $v0,40($fp)
lw $v1,68($fp)
slt $v0,$v1,$v0
beq $v0,$zero,$L58
b $L57
$L58:
lw $v0,40($fp)
lw $v1,68($fp)
slt $v0,$v0,$v1
beq $v0,$zero,$L59
lw $v0,40($fp)
sll $v1,$v0,2
lw $v0,72($fp)
addu $v0,$v1,$v0
lw $v0,0($v0)
sw $v0,24($fp)
b $L60
$L59:
lw $v0,64($fp)
lw $v1,68($fp)
slt $v0,$v0,$v1
beq $v0,$zero,$L61
b $L57
$L61:
lw $v0,28($fp)
sw $v0,24($fp)
lw $a0,24($fp)
la $a1,$LC8
la $t9,fopen
jal $ra,$t9
sw $v0,36($fp)
la $a0, __sF
la $t9,fgetc
jal $ra,$t9
sb $v0,32($fp)
$L62:
lb $v1,32($fp)
li $v0,-1 # 0xffffffffffffffff
bne $v1,$v0,$L64
b $L63
$L64:

```

```

lb $v0,32($fp)
move $a0,$v0
lw $a1,36($fp)
la $t9,fputc
jal $ra,$t9
la $a0, __sF
la $t9,fgetc
jal $ra,$t9
sb $v0,32($fp)
b $L62
$L63:
lw $a0,36($fp)
la $t9,fclose
jal $ra,$t9
$L60:
lw $v0,24($fp)
bne $v0,$zero,$L65
la $t9,imprime_uso
jal $ra,$t9
li $a0,1 # 0x1
la $t9,exit
jal $ra,$t9
$L65:
lw $v0,64($fp)
addu $v1,$v0,1
lw $v0,68($fp)
slt $v0,$v1,$v0
beq $v0,$zero,$L66
la $a0,$LC9
lw $a1,24($fp)
la $t9,printf
jal $ra,$t9
$L66:
lw $v0,80($fp)
beq $v0,$zero,$L67
lw $a0,24($fp)
lw $a1,76($fp)
lw $a2,84($fp)
la $t9,readBytes
jal $ra,$t9
b $L68
$L67:
lw $a0,24($fp)
lw $a1,76($fp)
lw $a2,84($fp)
la $t9,readLines
jal $ra,$t9
$L68:
la $a0,$LC10
la $t9,printf

```

```

jal $ra,$t9
lw $v0,40($fp)
addu $v0,$v0,1
sw $v0,40($fp)
b $L56
$L57:
lw $v1,64($fp)
lw $v0,68($fp)
bne $v1,$v0,$L55
lw $a0,28($fp)
la $t9,remove
jal $ra,$t9
$L55:
move $sp,$fp
lw $ra,56($sp)
lw $fp,52($sp)
addu $sp,$sp,64
j $ra
.end definirLineas
.size definirLineas, .-definirLineas
.rdata
.align 2
$LC12:
.ascii "help\000"
.align 2
$LC13:
.ascii "bytes\000"
.align 2
$LC14:
.ascii "lines\000"
.align 2
$LC15:
.ascii "version\000"
.data
.align 2
$LC16:
.word $LC12
.word 0
.word 0
.word 104
.word $LC13
.word 1
.word 0
.word 99
.word $LC14
.word 1
.word 0
.word 110
.word $LC15
.word 0

```

```

.word 0
.word 86
.word 0
.word 0
.word 0
.word 0
.globl memcpy
.rdata
.align 2
$LC11:
.ascii "hc:n:V\000"
.text
.align 2
.globl main
.ent main
main:
.frame $fp,152,$ra # vars= 104, regs= 3/0, args= 24, extra= 8
.mask 0xd0000000,-8
.fmask 0x00000000,0
.set noreorder
.cpload $t9
.set reorder
subu $sp,$sp,152
.cprestore 24
sw $ra,144($sp)
sw $fp,140($sp)
sw $gp,136($sp)
move $fp,$sp
sw $a0,152($fp)
sw $a1,156($fp)
sw $zero,32($fp)
la $v0,$LC11
sw $v0,36($fp)
addu $v0,$fp,40
la $v1,$LC16
move $a0,$v0
move $a1,$v1
li $a2,80 # 0x50
la $t9,memcpy
jal $ra,$t9
sw $zero,120($fp)
li $v0,10 # 0xa
sw $v0,124($fp)
sw $zero,128($fp)
lw $v0,156($fp)
lw $v0,0($v0)
sw $v0,nombre_programa
$L71:
lw $v1,32($fp)
li $v0,-1 # 0xffffffffffffffff

```

```

bne $v1,$v0,$L73
b $L72
$L73:
addu $v0,$fp,40
sw $zero,16($sp)
lw $a0,152($fp)
lw $a1,156($fp)
lw $a2,36($fp)
move $a3,$v0
la $t9,getopt_long
jal $ra,$t9
sw $v0,32($fp)
lw $v0,32($fp)
sw $v0,132($fp)
li $v0,99 # 0x63
lw $v1,132($fp)
beq $v1,$v0,$L77
lw $v1,132($fp)
slt $v0,$v1,100
beq $v0,$zero,$L87
li $v0,86 # 0x56
lw $v1,132($fp)
beq $v1,$v0,$L76
lw $v1,132($fp)
slt $v0,$v1,87
beq $v0,$zero,$L85
li $v0,-1 # 0xffffffffffffffff
lw $v1,132($fp)
beq $v1,$v0,$L71
li $v0,63 # 0x3f
lw $v1,132($fp)
beq $v1,$v0,$L83
b $L85
$L87:
li $v0,104 # 0x68
lw $v1,132($fp)
beq $v1,$v0,$L75
li $v0,110 # 0x6e
lw $v1,132($fp)
beq $v1,$v0,$L80
b $L85
$L75:
la $t9,imprime_uso
jal $ra,$t9
move $a0,$zero
la $t9,exit
jal $ra,$t9
$L76:
la $t9,imprime_version
jal $ra,$t9

```



```

move $a0,$zero
la $t9,exit
jal $ra,$t9
$L77:
li $v0,1 # 0x1
sw $v0,120($fp)
lw $v0,optarg
lb $v1,0($v0)
li $v0,43 # 0x2b
bne $v1,$v0,$L78
lw $v0,optarg
addu $v0,$v0,1
move $a0,$v0
la $t9,atoi
jal $ra,$t9
sw $v0,124($fp)
li $v0,1 # 0x1
sw $v0,128($fp)
b $L71
$L78:
lw $a0,optarg
la $t9,atoi
jal $ra,$t9
sw $v0,124($fp)
b $L71
$L80:
sw $zero,120($fp)
lw $v0,optarg
lb $v1,0($v0)
li $v0,43 # 0x2b
bne $v1,$v0,$L81
lw $v0,optarg
addu $v0,$v0,1
move $a0,$v0
la $t9,atoi
jal $ra,$t9
sw $v0,124($fp)
li $v0,1 # 0x1
sw $v0,128($fp)
b $L71
$L81:
lw $a0,optarg
la $t9,atoi
jal $ra,$t9
sw $v0,124($fp)
b $L71
$L83:
la $t9,imprime_uso
jal $ra,$t9
li $a0,1 # 0x1

```

```

    la $t9,exit
    jal $ra,$t9
$L85:
    la $t9,abort
    jal $ra,$t9
$L72:
    lw $v0,120($fp)
    sw $v0,16($sp)
    lw $v0,128($fp)
    sw $v0,20($sp)
    lw $a0,optind
    lw $a1,152($fp)
    lw $a2,156($fp)
    lw $a3,124($fp)
    la $t9,definirLineas
    jal $ra,$t9
    move $v0,$zero
    move $sp,$fp
    lw $ra,144($sp)
    lw $fp,140($sp)
    addu $sp,$sp,152
    j $ra
.end main
.size main, .-main

.comm nombre_programa,4
.ident "GCC: (GNU) 3.3.3 (NetBSD nb3 20040520)"s

```