

Organización de Computadoras 66.20

Trabajo Práctico 0

Autor	Padron	Correo electrónico					
Quino López, Julián	94224	julian.quino2@gmail.com					
Del Carril, Manuel	100772	manueldelcarril@gmail.com					
Bobadilla Catalan, German	90123	bobadillagerman@gmail.com					



Facultad de Ingeniería

Universidad de Buenos Aires

Av. Paseo Colón 850 - C1063ACV Ciudad Autónoma de Buenos Aires - Rep. Argentina Tel: +54 (11) 4343-0893 / 4343-0092 http://www.fi.uba.ar

Historial de Revisiones

Fecha	Revisor	Detalle					
09/04/2019	-	Entrega del TP					

Resumen

El siguiente trabajo práctico tiene como objetivo familiarizarse con el emulador GXEmul, que emula la arquitectura MIPS32, para lograr tal propósito se escribe en lenguaje C dos programas que permitan convertir archivos de texto desde Windows hacia UNIX y viceversa.

1 Introducción

Los archivos de texto requieren de un carácter especial (o secuencia de caracteres) para indicar el fin de una línea. La codificación de este varía según el sistema operativo, lo que lleva a la incorrecta visualización de un archivo en un sistema operativo que fue creado en otro. Linux utiliza el salto de línea (\n) mientras que Windows utiliza el retorno del carro seguido del salto de línea (\r\n).

2 Desarrollo

El algoritmo propuesto por el grupo consiste en recorrer caracter por caracter hasta encontrar un \r , si el programa usado es dos2unix y el siguiente caracter es un \n se elimnara el \r , si en cambio se usa el programa unix2dos se agregara \r antes de un \n , esto se realiza ya sea desde el archivo o utilizando el stream leído por entrada standard.

2.1 Comandos para compilar y ejecutar el programa

Se puede compilar los programas con los siguientes comandos:

```
$ gcc unix2dos.c -o unix2dos
$ gcc dos2unix.c -o dos2unix
```

Y luego ejecutarlos con los comandos:

```
$ ./unix2dos -i input.txt -o output.txt
$ ./dos2unix -i input.txt -o output.txt
```

En caso de sólo querer especificar el archivo de entrada, debe ejecutarse, por ejemplo, de la siguiente manera:

```
$ ./unix2dos -i input.txt -o -
$ ./dos2unix -i input.txt -o -
```

Análogamente si se quiere ingresar un archivo de salida:

```
$ ./unix2dos -i - -o output.txt
$ ./dos2unix -i - -o output.txt
```

Es decir que con un guión medio indicamos que no se proporcionará un archivo para entrada/salida, acorde a lo que indica el enunciado.

2.2 Otros comandos

Pueden utilizarse comandos tales como help y version, de la siguiente forma:

```
$ ./unix2dos -h
$ ./dos2unix -h
$ ./unix2dos -V
$ ./dos2unix -V
```

2.3 Código fuente de unix2dos.c

```
#include <stdio.h>
#include <string.h>
#include <getopt.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#define ERROR -1
#define SALIDA_EXITOSA 0
/**
* Procesa el archivo de entrada o el stream ingresado por stdin
 * Oparam inputFile
 * @param outputFile
 * @return un c digo
*/
int processInput(FILE *inputFile, FILE *outputFile) {
    while((c=fgetc(inputFile))!=EOF){
        if(c=='\r'){
            if((c=fgetc(inputFile))=='\n'){
                fprintf(outputFile,"\r\n");
            }else{
                fprintf(outputFile,"\r");
            }
        }else if(c=='\n'){
                fprintf(outputFile,"\r\n");
        }
        else{
            fprintf(outputFile,"%c",c);
    if(fclose(inputFile) == EOF) {
        fprintf(stderr, "Error fclose: %s\n", strerror( errno ));
        return ERROR;
    if(outputFile != stdout){
        if(fclose(outputFile) == EOF) {
            fprintf(stderr, "Error fclose: %s\n", strerror( errno))
            return ERROR;
        }
    return SALIDA_EXITOSA;
}
int main(int argc, char *argv[]) {
    int option = 0;
    const char *short_opt = "i:o:hV";
    struct option long_opt[] = {
```

```
{"version", no_argument,
                                       NULL, 'V'},
        {"help",
                  no_argument,
                                        NULL, 'h'},
                   required_argument, NULL, 'i'},
        {"input",
        {"output", required_argument, NULL, 'o'},
                                        NULL, 0}
        {NULL, 0,
};
FILE *inputFile = NULL;
FILE *outputFile = NULL;
while ((option = getopt_long(argc, argv, short_opt, long_opt,
   NULL)) != -1) {
    switch (option) {
        case 'V':
            printf("TP #0 de la materia Organizaci n de
               Computadoras \n");
            printf("Alumnos: \n");
            printf("
                            Bobadilla Catalan German\n
                Carril Manuel \n Quino Lopez Julian \n");
            return 0;
        case 'h':
            printf("Usage: \n");
            printf("
                            %s -h \n", argv[0]);
            printf("
                            %s -V \n", argv[0]);
            printf("
                            %s [options] \n", argv[0]);
            printf("Options: \n");
                            -V, --version Print version and
            printf("
               quit. \n");
            printf("
                            -h, --help
                                            Print this
               information. \n");
                            -o, --output
                                           Location of the
               output file. \n");
                            -i, --input
            printf("
                                          Location of the
               input file. \n");
            return 0;
        case 'i':
            if (strcmp(optarg, "-") != 0){
                    inputFile = fopen(optarg, "r");
                    if(inputFile == NULL){
                             fprintf(stderr, "Error archivo
                                entrada: %s\n", strerror(errno))
                            return ERROR;
                    }
            }
            break;
        case 'o':
            if(strcmp(optarg, "-") != 0){
                    outputFile = fopen(optarg, "w+");
                    if(outputFile == NULL) {
                            {\tt fprintf(stderr\,,\ "Error\ archivo}
                                salida: %s\n", strerror(errno));
                            return ERROR;
                    }
            }
```

```
break;
            default:
                // asi esta en el manual de getopt
                abort();
        }
    }
    if(inputFile == NULL) {
        inputFile = stdin;
    if(outputFile == NULL) {
        outputFile = stdout;
    if(processInput(inputFile, outputFile) == ERROR) {
        return ERROR;
    return SALIDA_EXITOSA;
}
2.4
    Código fuente de dos2unix.c
#include <stdio.h>
#include <string.h>
#include <getopt.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#define ERROR -1
#define SALIDA_EXITOSA 0
/**
* Procesa el archivo de entrada o el stream ingresado por stdin
 * @param inputFile
 * @param outputFile
 * @return un c digo
int processInput(FILE *inputFile, FILE *outputFile) {
    int c;
    while((c=fgetc(inputFile))!=EOF){
        if(c=='\r'){
                if((c=fgetc(inputFile))=='\n'){
                         fprintf(outputFile,"\n");
                else{
                         fprintf(outputFile,"\r");
                         fprintf(outputFile,"%c",c);
                }
        }
        else{
            fprintf(outputFile,"%c",c);
```

```
}
    }
    if(fclose(inputFile) == EOF) {
        fprintf(stderr, "Error fclose: %s\n", strerror( errno ));
        return ERROR;
    if(outputFile != stdout){
        if(fclose(outputFile) == EOF) {
            fprintf(stderr, "Error fclose: %s\n", strerror( errno))
            return ERROR;
        }
    }
    return SALIDA_EXITOSA;
}
int main(int argc, char *argv[]) {
    int option = 0;
    const char *short_opt = "i:o:hV";
    struct option long_opt[] = {
                                            NULL, 'V'},
            {\tt "version", no\_argument,}
            {"help", no_argument,
                                            NULL, 'h'},
            {"input",
                       required_argument, NULL, 'i'},
            {"output", required_argument, NULL, 'o'},
                                            NULL, 0}
            {NULL, 0,
    };
    FILE *inputFile = NULL;
    FILE *outputFile = NULL;
    while ((option = getopt_long(argc, argv, short_opt, long_opt,
       NULL)) != -1) {
        switch (option) {
            case 'V':
                printf("TP #0 de la materia Organizaci n de
                   Computadoras \n");
                printf("Alumnos: \n");
                                 Bobadilla Catalan German\n
                printf("
                    Carril Manuel \n Quino Lopez Julian \n");
                return 0;
            case 'h':
                printf("Usage: \n");
                printf("
                                %s -h \n", argv[0]);
                printf("
                                %s -V \n", argv[0]);
                printf("
                                %s [options] \n", argv[0]);
                printf("Options: \n");
                printf("
                                -V, --version Print version and
                   quit. \n");
                printf("
                                -h, --help
                                                Print this
                   information. \n");
                                -o, --output
                printf("
                                                Location of the
                   output file. \n");
                                -i, --input
                printf("
                                                Location of the
                   input file. \n");
                return 0;
```

```
case 'i':
                if(strcmp(optarg, "-") != 0){
                         inputFile = fopen(optarg, "r");
                         if(inputFile == NULL) {
                                 fprintf(stderr, "Error archivo
                                    entrada: %s\n", strerror(errno))
                                 return ERROR;
                         }
                }
                break;
            case 'o':
                if(strcmp(optarg, "-") != 0){
                         outputFile = fopen(optarg, "w+");
                         if(outputFile == NULL) {
                                 fprintf(stderr, "Error archivo
                                    salida: %s\n", strerror(errno));
                                 return ERROR;
                         }
                }
                break;
            default:
                // asi esta en el manual de getopt
                abort();
        }
    }
    if(inputFile == NULL) {
        inputFile = stdin;
    if(outputFile == NULL) {
        outputFile = stdout;
    if(processInput(inputFile, outputFile) == ERROR) {
        return ERROR;
    return SALIDA_EXITOSA;
}
```

3 Casos de prueba

A continuación se muestran unos casos de prueba desde la consola del GXEmul.

```
julian@julian-System: ~/Escritorio/Carpeta sin título/gxemul-6620-20070927
root@:~/66.20/TP0# od
0000000 S o l
                             -t c casos-de-prueba/entrada/docLinux
o d e s p u e s
              S
                    o
d
                                                    s
r
                                                          p
e
                                                               u
0000020
0000040
                                                                         \n
                                                                                h
0000060
                                                               d
d
                                                                                                e
0
s
                               s
                                          u
                                               e
                                                    s
d
                                                                     e
                                                                                     а
0
                                                                                           d
d
0000100
                                          e
                                                                     0
0000120
                                                               e
r
                                                                                e
h
             \n
                                                    у
                                                                         \n
0000140
                              Ь
              г
0000160
0000200
             \n
0000201
root@:~/66.20/TPO# ./unix2dos -i casos-de-prueba/entrada/docLinux -o casos-de-prue
ba/salida/docLinux-aplicarle-unix2dos
root@:~/66.20/TPO# od -t c casos-de-prueba/salida/docLinux-aplicarle-unix2dos
0000000 S o l o d e s p u e s d e l
                                                         p
e
                                                                                d
0000020
                    d
0000040
0000060
                                                    i
                                                                                                l
b
d
                                          u
                                               C
U
                                                          t
                                                                              \n
                                                                                           0
                                                                                     ĥ
              0
                                    s
                                                                          e
                                                                                          а
                                          P
P
c
i
                                                    e
0000100
                                               e
b
0000120
                   \г
                                                               у
                                                                                         \n
0000140
                    e
                                                                     P
U
                                                                          a
i
                                                                                     а
                                                                                                h
0000160
                                                                q
              а
                                               u
                                                    а
0000200
                         а
0000205
root@:~/66.20/TP0#
```

Figure 1: Prueba de transformar un archivo UNIX a Windows, utilizando archivo de entrada y salida.

8 0	julian(@julia	n-Sy	stem	: ~/Es	crito	orio/C	arpet	a sin	títul	o/gxe	emul-	6620	-200	7092	7	
root@:~/	66.20	/TP0	# od	-t	c ca	sos-	de-p	rueb	a/ent	trad	a/do	cLin	ux				
9000000	S	0	ı	0		d	e	S	P	u	e	S		d	e	ι	
9000020		d	e	S	a	S	t	Γ	e		Р	0	d	e	m	0	
9000040	S		Γ	e	S	u	C	i	t	а	Γ	\n	S	0	ι	0	
9000060		d	e	S	P	u	e	S		d	e		h	a	Ь	e	
9000100	Г	ι	0		P	e	Г	d	i	d	0		t	0	d	0	
0000120	/n	d	i	C	e		Т	У	ι	e	Γ	/n	e	Γ	e	s	
9000140		ι	i	Ь	Г	e		Р	а	Γ	a		h	a	C	e	
0000160	Г		C	u	a	ι	q	u	i	e	Γ		C	0	S	a	
9000200	/n																
9000201																	
root@:~/	66.20	/TP0	# ./	unix	(2dos	-i	caso	s - de	-pru	eba/	entr	ada/	docL	inux	-0	- 0	od -t c
9000000	S	0	ι	0		d	e	S	P	u	e	S		d	e	ι	
9000020		d	e	S	a	S	t	Γ	e		P	0	d	e	M	0	
9000040	S		Γ	e	S	u	C	i	t	a	Γ	\r	\n	S	0	ι	
9000060	0		d	e	S	Р	u	e	S		d	e		h	а	Ь	
0000100	е	Γ	ι	0		Р	e	Γ	d	i	d	0		t	0	d	
9000120	0	\r	\n	d	i	C	e		Т	У	ι	e	Γ	\г	\n	e	
0000140	Г	e	S		l	i	Ь	Γ	e		P	а	Γ	а		h	
0000160	а	C	e	Γ		C	u	a	ι	q	u	i	e	Γ		C	
9000200	0	S	а	\r	\n												
0000205 root@:~/																	

Figure 2: Prueba de transformar un arhivo UNIX a Windows, utilizando solamente archivo de entrada.

```
julian@julian-System: ~/Escritorio/Carpeta sin título/gxemul-6620-20070927
                         -t c casos-de-prueba/entrada/docWin.txt
root@:~/66.20/TP0# od
0000000
                e
                              В
0000020
            а
                                            n
                                                0
                                                                       ι
                                                                                g
e
0000040
                                                t
                                                         e
                                                                       у
г
                                                                           c
                                   а
                                       e
                                            s
                                                                           g
a
0000060
                              а
                                                     u
                                                              c
                                                                                0
                     e
                                   m
0000100
                              М
               \n
                     y
M
0000120
            а
                         e 361
                                            u
0000131
root@:~/
         66.20/TP0#
                                                                                - | od -t c
                                     casos-de-prueba/entrada/docWin.txt
                                                                             -0
                                                                  n
l
                                                     s
r
e
0000000
            S
                 e
                              В
0000020
                     s
                                       0
                                            n
t
                                                о
г
                                                        /n
                                                              e
P
                                                                           g
e
                                                                                ľ
0000040
                                                                  у
г
                                                                       C
                n
                                   e
                                       S
                         m
0000060
                 e
                                                                       g
                                                                               \n
0000100
                            361
0000120
                 e 361
                          i
                              q
                                   u
                                       e
0000127
root@:~/66.20/TP0#
```

Figure 3: Prueba de transformar un archivo Windows a UNIX, utilizando solo archivo de entrada.

```
julian@julian-System: ~/Escritorio/Carpeta sin título/gxemul-6620-20070927
root@:~/66.20/TPO# od -t c casos-de-prueba/entrada/docWin.txt
                                                      s
r
                                                                                  m
0000020
                                                                     e
P
                                                               \n
                                                                                   g
e
o
                                                                              c
g
a
0000040
                                                                c
0000060
                 1
                      e
                               a
M
                                                  S
                                                      u
                                                                    a
a
0000100
                                                  q
e
           \٢
                \n
0000120
                          e 361
                                         q
0000131
                     ./unix2dos
r Β
root@:~/66.20/TP0#
                                      casos-de
                                                -prueba/entrada/docWin.txt -o
                                                                                     | od -t c
0000000
                                                                              a
                                                                                  m
             S
                 e
                                        г
                                                  i
                                                      s
                                                           t
                                                                a
                                                                    n
0000020
                                                                    e
P
                                                                                   g
e
o
0000040
                                                  t
                      n
                               m
a
                                    а
                                         e
                                             S
                                                           e
                                                                         у
г
                                                                              g
a
                                                                c
0000060
                                                      u
                                                                    a
                                    m
                                                  s
0000100
           \٢
                \n
                                    e
i
                                                  q
e
0000120
                          e 361
0000131
root@:~/66.20/TP0#
```

Figure 4: Prueba de transformar un archivo Windows a un archivo Windows, la salida es la misma.

4 Código MIPS generado

4.1 Código fuente assembly de unix2dos.S

```
.file
                1 "unix2dos.c"
        .section .mdebug.abi32
        .previous
        .abicalls
        .rdata
        .align 2
$LCO:
        .ascii "\r\n\000"
        .align
$LC1:
        .ascii
                "\r\000"
        .align
$LC2:
                "%c\000"
        .ascii
        .align 2
$LC3:
                "Error fclose: %s\n\000"
        .ascii
        .text
        .align 2
        .globl processInput
                processInput
        .ent
processInput:
        .frame $fp,48,$31
                                         # vars= 8, regs= 3/0, args=
            16, extra= 8
        .mask 0xd0000000,-8
        .fmask 0x0000000,0
        .set
               noreorder
        .cpload $25
        .set
                reorder
        subu
                $sp,$sp,48
        .cprestore 16
                $31,40($sp)
        sw
                $fp,36($sp)
                $28,32($sp)
        SW
                $fp,$sp
        move
                $4,48($fp)
        sw
                $5,52($fp)
        sw
$L18:
        lw
                $4,48($fp)
                $25,fgetc
        la
        jal
                $31,$25
                $2,24($fp)
        sw
                $3,24($fp)
        lw
                                          # Oxffffffffffffff
                $2,-1
        li
        bne
                $3,$2,$L20
                $L19
        b
$L20:
        lw
                $3,24($fp)
        li
                $2,13
                                          # 0xd
```

```
bne
                 $3,$2,$L21
        lw
                 $4,48($fp)
        la
                 $25,fgetc
        jal
                 $31,$25
                 $2,24($fp)
        sw
                 $3,24($fp)
        lw
        li
                 $2,10
                                            # 0xa
        bne
                 $3,$2,$L22
                 $4,52($fp)
        lw
                 $5,$LCO
        la
                 $25,fprintf
        la
        jal
                 $31,$25
                 $L18
$L22:
                 $4,52($fp)
        lw
        la
                 $5,$LC1
        la
                 $25,fprintf
                 $31,$25
        jal
                 $L18
        b
$L21:
        lw
                 $3,24($fp)
        li
                                            # 0xa
                 $2,10
        bne
                 $3,$2,$L25
                 $4,52($fp)
        lw
                 $5,$LCO
        la
                 $25,fprintf
        la
        jal
                 $31,$25
                 $L18
$L25:
                 $4,52($fp)
        lw
        la
                 $5,$LC2
                 $6,24($fp)
        lw
                 $25,fprintf
        la
                 $31,$25
        jal
        b
                 $L18
$L19:
                 $4,48($fp)
        lw
        la
                 $25,fclose
        jal
                 $31,$25
                 $3,$2
        move
                 $2,-1
                                            # Oxfffffffffffffff
        li
        bne
                 $3,$2,$L27
        la
                 $25,__errno
        jal
                 $31,$25
                 $4,0($2)
        lw
                 $25, strerror
        la
                 $31,$25
        jal
        la
                 $4,_{sF}+176
        la
                 $5,$LC3
        move
                 $6,$2
        la
                 $25,fprintf
                 $31,$25
        jal
                 $2,-1
                                            # Oxfffffffffffffff
        li
                 $2,28($fp)
        sw
```

```
$L17
        b
$L27:
                 $3,52($fp)
        lw
        la
                 $2,_{sF}+88
        beq
                 $3,$2,$L28
                 $4,52($fp)
        lw
                 $25,fclose
        la
        jal
                 $31,$25
                 $3,$2
        move
                 $2,-1
                                           # Oxfffffffffffffff
        li
                 $3,$2,$L28
        bne
        la
                 $25,__errno
        jal
                 $31,$25
                 $4,0($2)
        lw
        la
                 $25, strerror
        jal
                 $31,$25
        la
                 $4,_{-}sF+176
                 $5,$LC3
        la
        move
                 $6,$2
                 $25,fprintf
        la
                 $31,$25
        jal
                                            # Oxffffffffffffff
        li
                 $2,-1
        sw
                 $2,28($fp)
                 $L17
        b
$L28:
                 $0,28($fp)
        sw
$L17:
        lw
                 $2,28($fp)
                 $sp,$fp
        move
                 $31,40($sp)
        lw
        lw
                 $fp,36($sp)
        addu
                 $sp,$sp,48
                 $31
        j
                 processInput
        .end
        .size
                 processInput, .-processInput
        .rdata
                 2
        .align
$LC5:
        .ascii
                 "version\000"
        .align
$LC6:
        .ascii
                 "help\000"
        .align
$LC7:
        .ascii
                 "input\000"
        .align
$LC8:
                 "output \000"
        .ascii
        .data
        .align
$LC9:
                 $LC5
        .word
        .word
                 0
        .word
```

```
.word
                86
                $LC6
        .word
        .word
                0
        .word
                0
               104
        .word
              $LC7
        .word
        .word
               1
        .word
        .word
              105
              $LC8
        .word
        .word
               1
        .word
        .word
              111
               0
        .word
        .word
        .word
        .word
               0
        .globl memcpy
        .rdata
        .align 2
$LC4:
        .ascii "i:o:hV\000"
        .align 2
$LC10:
        .ascii "TP #0 de la materia Organizaci\303\263n de
         Computadoras "
        .ascii "\n\000"
        .align 2
$LC11:
               "Alumnos: \n\000"
        .ascii
        .align 2
$LC12:
        .ascii
                "\tBobadilla Catalan German\n"
                "\tDel Carril Manuel \n"
        .ascii
        .ascii
                "\tQuino Lopez Julian \n\000"
        .align 2
$LC13:
        .ascii
                "Usage: \n\000"
        .align
$LC14:
                "\t%s -h \n\000"
        .ascii
        .align
$LC15:
        .ascii
                "\t%s -V \n\000"
        .align
$LC16:
                "\t%s [options] \n\000"
        .ascii
        .align
$LC17:
        .ascii
                "Options: \n\000"
        .align
                2
$LC18:
                "\t-V, --version Print version and quit. \n\000"
        .ascii
        .align
```

```
$LC19:
        .ascii "\t-h, --help
                                  Print this information. \n\000"
        .align 2
$LC20:
        .ascii "\t-o, --output
                                  Location of the output file. \n
          \000"
        .align 2
$LC21:
        .ascii "\t-i, --input
                                 Location of the input file. \n
           \000"
        .align 2
$LC22:
        .ascii "-\000"
        .align 2
$LC23:
        .ascii
               "r\000"
        .align
$LC24:
        .ascii
                "Error archivo entrada: %s\n\000"
        .align 2
$LC25:
               "w+\000"
        .ascii
        .align 2
$LC26:
        .ascii "Error archivo salida: %s\n\000"
        .text
        .align 2
        .globl main
        .ent
               main
main:
        .frame $fp,152,$31
                                       # vars= 104, regs= 3/0,
           args= 24, extra= 8
        .mask 0xd0000000,-8
        .fmask 0x0000000,0
              noreorder
        .set
        .cpload $25
        .set
             reorder
                $sp,$sp,152
        .cprestore 24
                $31,144($sp)
        SW
        sw
                $fp,140($sp)
                $28,136($sp)
        sw
       move
                $fp,$sp
                $4,152($fp)
        SW
                $5,156($fp)
        SW
        sw
                $0,32($fp)
                $2,$LC4
       la
                $2,36($fp)
        sw
               $2,$fp,40
        addu
                $3,$LC9
       la
               $4,$2
       move
       move
               $5,$3
                $6,80
                                       # 0x50
        li
        la
                $25, memcpy
```

```
$31,$25
         jal
                 $0,120($fp)
         sw
                  $0,124($fp)
         sw
$L31:
                 $2,$fp,40
         addu
                 $0,16($sp)
         sw
                 $4,152($fp)
        lw
        lw
                 $5,156($fp)
                 $6,36($fp)
        lw
                 $7,$2
        move
        la
                 $25,getopt_long
         jal
                 $31,$25
                 $2,32($fp)
         sw
                 $3,32($fp)
        lw
                                            # Oxffffffffffffff
        li
                 $2,-1
        bne
                 $3,$2,$L33
        b
                  $L32
$L33:
                  $2,32($fp)
         lw
                 $2,132($fp)
         sw
        li
                 $2,104
                                            # 0x68
        lw
                 $3,132($fp)
        beq
                 $3,$2,$L36
                 $3,132($fp)
        lw
                 $2,$3,105
         slt
                 $2,$0,$L45
        beq
        li
                 $2,86
                                            # 0x56
        lw
                 $3,132($fp)
                 $3,$2,$L35
        beq
                  $L43
        b
$L45:
        li
                 $2,105
                                            # 0x69
        lw
                 $3,132($fp)
                 $3,$2,$L37
        beq
                                            # 0x6f
        li
                 $2,111
        lw
                 $3,132($fp)
        beq
                 $3,$2,$L40
        b
                  $L43
$L35:
                 $4,$LC10
         la
                 $25, printf
         la
         jal
                  $31,$25
                 $4,$LC11
         la
        la
                 $25, printf
                 $31,$25
         jal
                 $4,$LC12
        la
                 $25, printf
        la
                 $31,$25
         jal
                 $0,128($fp)
        sw
        b
                  $L30
$L36:
                 $4,$LC13
        la
         la
                 $25, printf
         jal
                 $31,$25
```

```
$2,156($fp)
lw
la
         $4,$LC14
         $5,0($2)
lw
la
         $25,printf
jal
         $31,$25
         $2,156($fp)
lw
         $4,$LC15
la
lw
         $5,0($2)
         $25,printf
la
         $31,$25
jal
         $2,156($fp)
lw
la
         $4,$LC16
         $5,0($2)
lw
la
         $25, printf
jal
         $31,$25
la
         $4,$LC17
la
         $25,printf
jal
         $31,$25
la
         $4,$LC18
la
         $25, printf
jal
         $31,$25
la
         $4,$LC19
la
         $25, printf
         $31,$25
jal
         $4,$LC20
la
la
         $25, printf
jal
         $31,$25
la
         $4,$LC21
         $25, printf
la
         $31,$25
jal
sw
         $0,128($fp)
b
         $L30
         $4,optarg
lw
la
         $5,$LC22
la
         $25,strcmp
         $31,$25
jal
beq
         $2,$0,$L31
         $4,optarg
lw
         $5,$LC23
la
la
         $25, fopen
jal
         $31,$25
         $2,120($fp)
sw
lw
         $2,120($fp)
         $2,$0,$L31
bne
la
         $25,__errno
         $31,$25
jal
         $4,0($2)
lw
         $25, strerror
la
jal
         $31,$25
la
         $4,_{sF}+176
         $5,$LC24
la
         $6,$2
move
la
         $25,fprintf
```

\$L37:

```
jal
                 $31,$25
        b
                 $L31
$L40:
        lw
                 $4,optarg
                 $5,$LC22
        la
                 $25,strcmp
        la
                 $31,$25
        jal
        beq
                 $2,$0,$L31
                 $4,optarg
        lw
                 $5,$LC25
        la
                 $25, fopen
        la
        jal
                 $31,$25
                 $2,124($fp)
        sw
                 $2,124($fp)
        lw
                 $2,$0,$L31
        bne
        la
                 $25,__errno
                 $31,$25
        jal
        lw
                 $4,0($2)
        la
                 $25, strerror
                 $31,$25
        jal
                 $4,_{-s}F+176
        la
        la
                 $5,$LC26
        move
                 $6,$2
                 $25,fprintf
        la
                 $31,$25
        jal
                 $2,-1
                                            # Oxfffffffffffffff
        li
        sw
                 $2,128($fp)
        b
                 $L30
$L43:
                 $25,abort
        la
        jal
                 $31,$25
$L32:
                 $2,120($fp)
        lw
                 $2,$0,$L46
        bne
        la
                 $2,__sF
                 $2,120($fp)
        sw
$L46:
        lw
                 $2,124($fp)
        bne
                 $2,$0,$L47
                 $2,_{sF}+88
        la
                 $2,124($fp)
        sw
$L47:
                 $4,120($fp)
        lw
        lw
                 $5,124($fp)
                 $25, processInput
        la
        jal
                 $31,$25
        move
                 $3,$2
                                            # Oxffffffffffffff
                 $2,-1
        li
                 $3,$2,$L48
        bne
                                            # Oxfffffffffffffff
        li
                 $3,-1
                 $3,128($fp)
        sw
                 $L30
        b
$L48:
                 $0,128($fp)
        sw
```

```
$L30:
        lw
                 $2,128($fp)
        move
                 $sp,$fp
        lw
                 $31,144($sp)
        lw
                 $fp,140($sp)
                 $sp,$sp,152
        addu
                 $31
        j
        .end
                 main
                 main, .-main
        .size
                 "GCC: (GNU) 3.3.3 (NetBSD nb3 20040520)"
        .ident
```

4.2 Código fuente assembly de dos2unix.S

```
1 "dos2unix.c"
        .file
        .section .mdebug.abi32
        .previous
        .abicalls
        .rdata
        .align
$LC0:
                 "\n\000"
        .ascii
        .align
$LC1:
                 "\r\000"
        .ascii
        .align
                 2
$LC2:
        .ascii
                 "%c\000"
        .align
                2
$LC3:
                 "Error fclose: %s\n\000"
        .ascii
        .text
        .align 2
        .globl processInput
        .ent
                processInput
processInput:
                                          # vars= 8, regs= 3/0, args=
        .frame $fp,48,$31
             16, extra= 8
                 0xd0000000,-8
        .mask
        .fmask 0x0000000,0
        .set
                 noreorder
        .cpload $25
        .set
                 reorder
        subu
                 $sp,$sp,48
        .cprestore 16
                 $31,40($sp)
        sw
                 $fp,36($sp)
        sw
                 $28,32($sp)
        SW
                 $fp,$sp
        move
                 $4,48($fp)
        sw
                 $5,52($fp)
        sw
$L18:
                 $4,48($fp)
        lw
                 $25,fgetc
        la
```

```
$31,$25
        jal
        sw
                 $2,24($fp)
                 $3,24($fp)
        lw
                                            # Oxffffffffffffff
        li
                 $2,-1
        bne
                 $3,$2,$L20
                 $L19
        b
$L20:
        lw
                 $3,24($fp)
                 $2,13
                                            # 0xd
        li
                 $3,$2,$L21
        bne
                 $4,48($fp)
        lw
        la
                 $25,fgetc
        jal
                 $31,$25
                 $2,24($fp)
        sw
        lw
                 $3,24($fp)
        li
                 $2,10
                                            # 0xa
        bne
                 $3,$2,$L22
        lw
                 $4,52($fp)
        la
                 $5,$LCO
        la
                 $25,fprintf
                 $31,$25
        jal
                 $L18
        b
$L22:
                 $4,52($fp)
        lw
                 $5,$LC1
        la
                 $25,fprintf
        la
        jal
                 $31,$25
        lw
                 $4,52($fp)
                 $5,$LC2
        la
                 $6,24($fp)
        lw
        la
                 $25,fprintf
                 $31,$25
        jal
                 $L18
        b
$L21:
                 $4,52($fp)
        lw
                 $5,$LC2
        la
        lw
                 $6,24($fp)
        la
                 $25,fprintf
                 $31,$25
        jal
                 $L18
        b
$L19:
        lw
                 $4,48($fp)
                 $25,fclose
        la
        jal
                 $31,$25
                 $3,$2
        move
                                            # Oxfffffffffffffff
        li
                 $2,-1
        bne
                 $3,$2,$L25
        la
                 $25,__errno
                 $31,$25
        jal
        lw
                 $4,0($2)
                 $25, strerror
        la
                 $31,$25
        jal
                 $4,_{sF}+176
        la
        la
                 $5,$LC3
```

```
move
                 $6,$2
                 $25,fprintf
        la
                 $31,$25
        jal
                                           # Oxffffffffffffff
        li
                 $2,-1
        sw
                 $2,28($fp)
                 $L17
        b
$L25:
        lw
                 $3,52($fp)
                 $2,_{sF}+88
        la
                 $3,$2,$L26
        beq
                 $4,52($fp)
        lw
        la
                 $25,fclose
        jal
                 $31,$25
                 $3,$2
        move
                                           # Oxfffffffffffffff
        li
                 $2,-1
        bne
                 $3,$2,$L26
        la
                 $25,__errno
                 $31,$25
        jal
        lw
                 $4,0($2)
                 $25, strerror
        la
        jal
                 $31,$25
        la
                 $4,_sF+176
        la
                 $5,$LC3
                 $6,$2
        move
                 $25,fprintf
        la
                 $31,$25
        jal
                                           # Oxfffffffffffffff
        li
                 $2,-1
                 $2,28($fp)
        sw
                 $L17
        b
$L26:
        sw
                 $0,28($fp)
$L17:
        lw
                 $2,28($fp)
        move
                 $sp,$fp
        lw
                 $31,40($sp)
        lw
                 $fp,36($sp)
        addu
                 $sp,$sp,48
                 $31
        j
                 processInput
        .end
                 processInput, .-processInput
        .size
        .rdata
        .align
$LC5:
                 "version\000"
        .ascii
        .align
$LC6:
        .ascii
                 "help\000"
        .align
$LC7:
                 "input\000"
        .ascii
        .align
$LC8:
        .ascii
                 "output \000"
        .data
```

```
.align 2
$LC9:
               $LC5
       .word
       .word
               0
               0
       .word
       .word
               86
             $LC6
       .word
       .word
       .word
               0
              104
        .word
        .word
               $LC7
       .word
               1
              0
       .word
       .word 105
       .word $LC8
       .word
              1
               0
       .word
              111
       .word
       .word
       .word
               0
       .word
               0
       .word
               0
       .globl memcpy
       .rdata
       .align
               2
$LC4:
       .ascii "i:o:hV\000"
       .align 2
$LC10:
        .ascii "TP #0 de la materia Organizaci\303\263n de
          Computadoras "
               "\n\000"
       .ascii
       .align 2
$LC11:
        .ascii
               "Alumnos: \n\000"
       .align 2
$LC12:
       .ascii
               "\tBobadilla Catalan Germann"
               "\tDel Carril Manuel \n"
       .ascii
               "\tQuino Lopez Julian \n\000"
       .ascii
       .align
$LC13:
       .ascii
               "Usage: \n\000"
       .align 2
$LC14:
               "\t%s -h \n\000"
       .ascii
       .align
$LC15:
               "\t%s -V \n\000"
        .ascii
       .align 2
$LC16:
       .ascii
               "\t%s [options] \n\000"
       .align 2
$LC17:
```

```
"Options: \n\000"
        .ascii
        .align 2
$LC18:
        .ascii
                "\t-V, --version Print version and quit. \n\000"
        .align 2
$LC19:
               "\t-h, --help
                                 Print this information. \n\000"
        .ascii
        .align 2
$LC20:
        .ascii "\t-o, --output
                                 Location of the output file. \n
         \000"
        .align 2
$LC21:
        .ascii "\t-i, --input
                                 Location of the input file. \n
          \000"
        .align 2
$LC22:
        .ascii "-\000"
        .align 2
$LC23:
               "r\000"
        .ascii
        .align 2
$LC24:
        .ascii "Error archivo entrada: %s\n\000"
        .align 2
$LC25:
        .ascii "w+\000"
        .align 2
$LC26:
        .ascii "Error archivo salida: %s\n\000"
        .text
        .align 2
        .globl main
        .ent
               main
main:
                                       # vars= 104, regs= 3/0,
        .frame $fp,152,$31
           args= 24, extra= 8
        .mask 0xd0000000,-8
        .fmask 0x0000000,0
               noreorder
        .set
        .cpload $25
        .set
               reorder
        subu
                $sp,$sp,152
        .cprestore 24
               $31,144($sp)
        SW
        sw
                $fp,140($sp)
               $28,136($sp)
       SW
               $fp,$sp
       move
                $4,152($fp)
        sw
                $5,156($fp)
        SW
                $0,32($fp)
        SW
               $2,$LC4
        la
                $2,36($fp)
        SW
        addu
               $2,$fp,40
```

```
$3,$LC9
        la
                 $4,$2
        move
                 $5,$3
        move
        li
                 $6,80
                                            # 0x50
                 $25, memcpy
        la
         jal
                 $31,$25
                 $0,120($fp)
         sw
                 $0,124($fp)
         sw
$L29:
                 $2,$fp,40
         addu
                 $0,16($sp)
         sw
        lw
                 $4,152($fp)
                 $5,156($fp)
        lw
        lw
                 $6,36($fp)
        move
                 $7,$2
                 $25,getopt_long
        la
         jal
                 $31,$25
                 $2,32($fp)
         SW
        lw
                 $3,32($fp)
        li
                                            # Oxffffffffffffff
                 $2,-1
        bne
                 $3,$2,$L31
        b
                 $L30
$L31:
                 $2,32($fp)
        lw
                 $2,132($fp)
         sw
                                            # 0x68
        li
                 $2,104
        lw
                 $3,132($fp)
                 $3,$2,$L34
        beq
                 $3,132($fp)
        lw
                 $2,$3,105
         slt
        beq
                 $2,$0,$L43
                                            # 0x56
        li
                 $2,86
        lw
                 $3,132($fp)
                 $3,$2,$L33
        beq
        b
                  $L41
$L43:
                 $2,105
                                            # 0x69
        li
        lw
                  $3,132($fp)
                 $3,$2,$L35
         beq
                                            # 0x6f
         li
                 $2,111
         lw
                 $3,132($fp)
         beq
                 $3,$2,$L38
                  $L41
$L33:
                 $4,$LC10
        la
         la
                 $25, printf
                 $31,$25
         jal
                 $4,$LC11
        la
                 $25, printf
        la
         jal
                 $31,$25
                 $4,$LC12
        la
        la
                 $25, printf
                 $31,$25
         jal
         sw
                  $0,128($fp)
```

```
$L28
         b
$L34:
                  $4,$LC13
         la
         la
                  $25,printf
         jal
                  $31,$25
                  $2,156($fp)
         lw
         la
                  $4,$LC14
         lw
                  $5,0($2)
         la
                  $25, printf
                  $31,$25
         jal
                  $2,156($fp)
         lw
         la
                  $4,$LC15
         lw
                  $5,0($2)
                  $25, printf
         la
         jal
                  $31,$25
         lw
                  $2,156($fp)
         la
                  $4,$LC16
         lw
                  $5,0($2)
         la
                  $25, printf
         jal
                  $31,$25
                  $4,$LC17
         la
         la
                  $25, printf
         jal
                  $31,$25
                  $4,$LC18
         la
                  $25, printf
         la
                  $31,$25
         jal
         la
                  $4,$LC19
         la
                  $25, printf
                  $31,$25
         jal
                  $4,$LC20
         la
         la
                  $25, printf
                  $31,$25
         jal
                  $4,$LC21
         la
                  $25, printf
         la
         jal
                  $31,$25
                  $0,128($fp)
         sw
                  $L28
         b
$L35:
         lw
                  $4,optarg
                  $5,$LC22
         la
         la
                  $25, strcmp
         jal
                  $31,$25
                  $2,$0,$L29
         beq
         lw
                  $4,optarg
                  $5,$LC23
         la
         la
                  $25, fopen
                  $31,$25
         jal
                  $2,120($fp)
         sw
                  $2,120($fp)
         lw
         bne
                  $2,$0,$L29
         la
                  $25,__errno
         jal
                  $31,$25
                  $4,0($2)
         lw
         la
                  $25, strerror
```

```
jal
                 $31,$25
        la
                 $4,__sF+176
                 $5,$LC24
        la
        move
                 $6,$2
        la
                 $25,fprintf
                 $31,$25
        jal
                 $L29
        b
$L38:
        lw
                 $4,optarg
                 $5,$LC22
        la
                 $25,strcmp
        la
        jal
                 $31,$25
        beq
                 $2,$0,$L29
                 $4,optarg
        lw
        la
                 $5,$LC25
        la
                 $25, fopen
        jal
                 $31,$25
                 $2,124($fp)
        SW
        lw
                 $2,124($fp)
                 $2,$0,$L29
        bne
        la
                 $25,__errno
        jal
                 $31,$25
        lw
                 $4,0($2)
                 $25, strerror
        la
                 $31,$25
        jal
                 $4,__sF+176
        la
        la
                 $5,$LC26
        move
                 $6,$2
                 $25,fprintf
        la
                 $31,$25
        jal
                                            # Oxfffffffffffffff
        li
                 $2,-1
        sw
                 $2,128($fp)
        b
                 $L28
$L41:
        la
                 $25, abort
        jal
                 $31,$25
$L30:
        lw
                 $2,120($fp)
        bne
                 $2,$0,$L44
                 $2,__sF
        la
                 $2,120($fp)
        sw
$L44:
                 $2,124($fp)
        lw
                 $2,$0,$L45
        bne
        la
                 $2,__sF+88
                 $2,124($fp)
        sw
$L45:
                 $4,120($fp)
        lw
                 $5,124($fp)
        lw
        la
                 $25, processInput
        jal
                 $31,$25
                 $3,$2
        move
        li
                 $2,-1
                                            # Oxfffffffffffffff
                 $3,$2,$L46
        bne
```

```
$3,-1
                                           # Oxffffffffffffff
        li
                 $3,128($fp)
        SW
                 $L28
$L46:
                 $0,128($fp)
        SW
$L28:
                 $2,128($fp)
        lw
                 $sp,$fp
        move
                 $31,144($sp)
        lw
                 $fp,140($sp)
        lw
        addu
                 $sp,$sp,152
                 $31
        j
        .end
                 main
        .size
                 main, .-main
                 "GCC: (GNU) 3.3.3 (NetBSD nb3 20040520)"
        .ident
```

5 Conclusiones

El trabajo práctico nos resultó interesante, no por el programa a desarrollar en sí, sino por lo que representó trabajar con el emulador GXEmul, emular la arquitectura MIPS, crear el túnel de comunicación entre el host OS (Linux, distribución Ubuntu) y el guest OS (NetBSD). Aprendimos como transferir archivos entre ambos sistemas y también ciertas cuestiones del lenguaje C con el cual no estábamos toalmente familiarizados.

References

- [1] GetOpt library, https://www.gnu.org/software/libc/manual/html $_node/Example-of-Getopt.html$.
- [2] StackOverflow, https://www.stackoverflow.com.