FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Trabajo Práctico Nº 1

Infraestructura básica

Organización de Computadoras

2º Cuatrimestre 2018

participantes	nº padrón	mail
CAFERATTA, Gianmarco	99423	giancaferatta@gmail.com
KELMAN, Uriel	99616	urielkelman@gmail.com
MARIJUAN, Magalí	100070	maguimar001@gmail.com

Detalles de implementación

Para lograr una mayor división del trabajo y con el objetivo de testear mejor las distintas funcionalidades que fueron necesario programar para la elaboración del tp, se decidió dividir al código en distintas librerías e incluir los distintos archivos *.h* en caso de necesitarlos.
La primera librería escrita fue *bit_library.h*. En esta, se pueden encontrar distintas funciones que permiten realizar operaciones con bits, desde encontrar si un bit dentro de un byte está encendido o no hasta elaborar máscaras determinadas. Esta librería fue codificada sabiendo que las funciones que provee serían necesarias para realizar las operaciones necesarias para realizar el *encoding* y *decoding*.
En segundo lugar, nos encontramos con la librería *base_64.h* Esta librería es la que servirá como interfaz para el programa principal. En ella, se encuentran las funciones encargadas de realizar el *encoding* y el *decoding* de los distintos bytes que se pasen por parámetro.
Por último, tenemos el problema principal. En la función *main*() hay un ciclo que recorre el vector de parámetros que se recibe al ejecutar el programa, y los va procesando hasta terminar. En caso de que haya algún error en el pasaje de parámetros, se imprime un error por **stderr**. Los errores están relacionados a un mal orden en los parámetros, a la utilización de nombres incorrectos y también al pasaje de una cantidad incorrecta de los mismos. Luego, en el programa hay dos funciones que se encargan de encodificar o decodificar un archivo, que son las que se encargan de leer el archivo de entrada, encodificar o decodificar según corresponda, y escribir sobre el archivo de salida.

Comandos de compilación

El comando utizado para compilar fue:

```
gcc -g -std=c99 -o main *.c
```

gcc es el compilador utilzado.

$-g$ es un flag que indica al compilador que el ejecutable debe incluir los ficheros correspondientes al código fuente. Esta información fue utilizada a la hora de realizar depuraciones del programa con *gdb*.

$-std = c99$ activa el estándar C99. Se usó por tener un mayor conocimiento respecto al estándar y lograr una mayor comodidad, por ejemplo para inicializar variables al momento de declarar un ciclo.

$-O0$, este flag fue utilizado para apagar las optimizaciones que por default intenta realizar el compilador, para lograr que el código assembly generado sea más comprensible.

$-o$ indica el nombre del archivo ejecutable que será el resultado de la compilación. Nótese que no fueron especificado todos los nombres de los archivos, sino que se utilizó *.C, lo que genera la compilación del main y de todas las dependecias recursivamente.

Corridas de prueba

A continuación se detallan las corridas de prueba del programa. Se intentó que éstas sean lo más abarcativas posibles, mostrando casos en los que funciona el encoding y el decoding, tomando archivos reales físicos y también los archivos estándar de entrada y salida, así como también se intenta mostrar cuáles son todos los posibles errores por los cuáles el programa muestra un error.
Por otro lado, hay que aclara que para cada librería fueron programadas las pruebas pertinentes. Son pruebas de tipo unitarias que prueban varios casos de las funcionalidades programadas en las librerías.

Empecemos entonces por algunas corridas triviales por *stdin*, para mostrar que se codifica y decodifica correctamente.
Codificamos el caracter ASCII M:

```
echo -n M | ./main
TQ==
```

Codificamos los caracteres ASCII M y a:

```
echo -n Ma |./main
TWE=
```

Codificamos ahora la cadena de 3 bytes Man:

```
echo -n Man |./main
TWFu
```

Codificamos una cadena de 4 bytes (recordar que cada tres bytes, el resultado del encoding tendrá longitud 4):

```
echo -n ManM |./main
TWFuTQ==
```

Probamos codificar sin especificar ninguna acción y vemos el error resultante:

```
echo -n Man |./main -a
Please, specify the action. Default action is encoding.
```

O también, podemos ver que pasa si intentamos pasar como parámetro una acción inexistente:

```
echo -n Man |./main -a stream
Invalid action. Please use an allowed action. See the help menu.
```

Ahora, veamos que sucede cuando encodeamos y luego desencodeamos:

```
echo -n Man |./main | ./main -a decode
Man
```

Y un caso de prueba más encodeando y desencodeando 4 caracteres:

```
echo -n ManM |./main | ./main -a decode
ManM
```

Un posible escenario de error sería intentar decodificar 5 caracteres:

```
echo -n "htjdW" | ./main -a decode
An error ocurred while trying to perform your request.
```

Probemos ahora qué sucede cuando especificamos archivos de entrada y salida.
Veamos un caso de error, donde no se especifica el archivo de entrada.

```
./main -a encode -i
Missing input file. Try again.
```

O también, un caso donde no existe el archivo que se quiere encodear.

```
./main -a encode -i entrada.txt
Invalid file. Please use a file that exist or check the path.
```

Otro caso de error sería no especificar el archivo de salida (nótese que fue creado el archivo de entrada):

```
./main -a encode -i entrada.txt -o
Missing output file. Try again.
```

Ahora, vistos todos los posibles errores, veamos qué sucede colocando todos los parámetros correctamente:

```
echo -n "Man" > entrada.txt | ./main -a encode -i entrada.txt -o salida.txt | cat salida.txt
TWFu
```

Y decodificando ahora este archivo de salida:

```
./main -a decode -i salida.txt -o entrada.txt | cat entrada.txt
Man
```

Ahora probemos con un archivo más grande, de tamaño extenso:

```
echo -n "Siempre se ha arguido que el problema del capitalismo es que es materialista,
mientras que el colectivismo puede permitirse prestar atención a lo no material. Pero
la experiencia ha sido exactamente la contraria. No hay sociedades que hayan enfatizado
los requisitos materiales del bienestar como las colectivistas. Son las sociedades libres
las que, de lejos, han permitido un mayor desarrollo a los aspectos inmateriales,
espirituales, artisticos del bienestar." > entrada.txt | ./main -a encode -i entrada.txt
-o salida.txt | ./main -a decode -i salida.txt -o entrada.txt | cat entrada.txt
Siempre se ha arguido que el problema del capitalismo es que es materialista, mientras
que el colectivismo puede permitirse prestar atención a lo no material. Pero la
experiencia ha sido exactamente la contraria. No hay sociedades que hayan enfatizado
los requisitos materiales del bienestar como las colectivistas. Son las sociedades
libres las que, de lejos, han permitido un mayor desarrollo a los aspectos inmateriales,
espirituales, artisticos del bienestar.
```

Por último, generamos archivos de caracteres aleatorios y una longitud que va creciendo constantemente:

```
n=1;
while :; do
>head -c $n </dev/urandom >in.bin;
>./main -a encode -i in.bin -o out.b64;
>./main -a decode -i out.b64 -o out.bin;
>if diff in.bin out.bin; then :; else
>echo ERROR: $n;
>break;
>fi;
>echo ok: $n;
>n=$((n+1));
>done
ok: 1
ok: 2
ok: 3
ok: 4
ok: 5
ok: 6
ok: 7
ok: 8
ok: 9
ok: 10
ok: 11
  .
  .
  .
ok: 97
ok: 98
ok: 99
ok: 100
```

Y también probamos actualizando n de forma tal que crezca muchísimo más rápido y los archivos sean de mayor longitud:

```
n=1;
while :;
>do head -c $n </dev/urandom >in.bin;
>./main -a encode -i in.bin -o out.b64;
>./main -a decode -i out.b64 -o out.bin;
>if diff in.bin out.bin; then :; else >
echo ERROR: $n;
>break;
>fi; echo ok:
>$n;
>n=$((n*2));
>done
ok: 1
ok: 2
ok: 4
ok: 8
ok: 16
ok: 32
ok: 64
ok: 128
  .
  .
  .
ok: 262144
ok: 524288
ok: 1048576
```

Código fuente

bit_library.h

```c
#include <stdlib.h>
#include <stdbool.h>
// Todos los parametros de posicion son bits y empiezan en 0

// Recibe un byte y una posicion y devuelve un booleano indicando si el byte es 1 o 0.
bool byte_nth_bit(char byte, int n);

// Recibe un byte y una posicion y hace una mascara de ese bit.
char byte_mask_nth_bit(char byte, int n);

// Recibe byte y dos numeros y devuelve todos los bits comprendidos en ese intervalo.
char byte_extract_n_bits(char byte, int initial_n, int last_n);
```

```
13
14 // Recibe una cadena de bytes y una posicion y toma el bit enesimo y lo
       devuelve como booleano.
15 bool bytes_nth_bit(char* bytes, int n);
16
17 // Recibe una cadena de bytes y del byte en donde se encuentra el bit enesimo,
       hace una mascara con ese bit.
18 char bytes_mask_nth_bit(char* bytes, int n);
19
20 // Recibe una cadena de bytes y devuelve un intervalo de bits.
21 // Se debe extraer almenos un byte.
22 char* bytes_extract_n_bits(char* bytes, int intial_n, int last_n);
```

bit_library.c

```c
#include <stdlib.h>
#include <stdbool.h>
#define IS_BIG_ENDIAN true

bool byte_nth_bit(char byte, int n){
  if(IS_BIG_ENDIAN) n = abs(n-7);
  char mask = 0x01;
  byte = byte >> n;
  return mask & byte;
}

char byte_mask_nth_bit(char byte, int n){
  char bit = byte_nth_bit(byte,n);
  if(IS_BIG_ENDIAN) n = abs(n-7);
  return bit << n;
}

char byte_extract_n_bits(char byte, int initial_n, int last_n){
  char retval = 0x00;
  for(int i=initial_n;i<=last_n;i++){
    retval = retval | byte_mask_nth_bit(byte, i);
  }
  return retval;
}

bool bytes_nth_bit(char* bytes, int n){
  int byte_location = n/8;
  int bit_location = n-byte_location*8;
  return byte_nth_bit(bytes[byte_location], bit_location);
}

char bytes_mask_nth_bit(char* bytes, int n){
  int byte_location = n/8;
  int bit_location = n-byte_location;
  return byte_mask_nth_bit(bytes[byte_location], bit_location);
}

char* bytes_extract_n_bits(char* bytes, int initial_n, int last_n){
  int initial_byte = initial_n / 8;
  int last_byte = last_n / 8;
  char* bits = (char*)malloc(last_byte - initial_byte+1);
  if(!bits) return NULL;
  for(int i=initial_byte;i<=last_byte;i++){
    int initial_bit = 0;
    if (i==initial_byte) initial_bit = initial_n%8;
    int last_bit=7;
    if (i==last_byte) last_bit = last_n%8;
    bits[i-initial_byte] = byte_extract_n_bits(bytes[i], initial_bit, last_bit)
    ;
  }
  return bits;
}
```

```c
#include "bit_library.h"
#include <stdio.h>
#include <stdbool.h>

void print_test(const char* msg, bool cond){
  if(cond) printf("%s ... %s", msg, "OK");
  else printf("%s ... %s", msg, "ERROR");
  printf("\n");
}

void test_byte_nth_bit(){
  printf("\n");
  printf("--- INICIO DE PRUEBAS DE BYTE_NTH_BIT ---\n");

  print_test("El bit 0 de 'a' es correcto", !byte_nth_bit('a',0));
  print_test("El bit 1 de 'a' es correcto", byte_nth_bit('a',1));
  print_test("El bit 2 de 'a' es correcto", byte_nth_bit('a',2));
  print_test("El bit 3 de 'a' es correcto", !byte_nth_bit('a',3));
  print_test("El bit 4 de 'a' es correcto", !byte_nth_bit('a',4));
  print_test("El bit 5 de 'a' es correcto", !byte_nth_bit('a',5));
  print_test("El bit 6 de 'a' es correcto", !byte_nth_bit('a',6));
  print_test("El bit 7 de 'a' es correcto", byte_nth_bit('a',7));

  print_test("El bit 0 de '*' es correcto", !byte_nth_bit('*',0));
  print_test("El bit 1 de '*' es correcto", !byte_nth_bit('*',1));
  print_test("El bit 2 de '*' es correcto", byte_nth_bit('*',2));
  print_test("El bit 3 de '*' es correcto", !byte_nth_bit('*',3));
  print_test("El bit 4 de '*' es correcto", byte_nth_bit('*',4));
  print_test("El bit 5 de '*' es correcto", !byte_nth_bit('*',5));
  print_test("El bit 6 de '*' es correcto", byte_nth_bit('*',6));
  print_test("El bit 7 de '*' es correcto", !byte_nth_bit('*',7));

  print_test("El bit 0 de 0xff es correcto", byte_nth_bit(0xff,0));
  print_test("El bit 1 de 0xff es correcto", byte_nth_bit(0xff,1));
  print_test("El bit 2 de 0xff es correcto", byte_nth_bit(0xff,2));
  print_test("El bit 3 de 0xff es correcto", byte_nth_bit(0xff,3));
  print_test("El bit 4 de 0xff es correcto", byte_nth_bit(0xff,4));
  print_test("El bit 5 de 0xff es correcto", byte_nth_bit(0xff,5));
  print_test("El bit 6 de 0xff es correcto", byte_nth_bit(0xff,6));
  print_test("El bit 7 de 0xff es correcto", byte_nth_bit(0xff,7));

  print_test("El bit 0 de 0x0f es correcto", !byte_nth_bit(0x0f,0));
  print_test("El bit 1 de 0x0f es correcto", !byte_nth_bit(0x0f,1));
  print_test("El bit 2 de 0x0f es correcto", !byte_nth_bit(0x0f,2));
  print_test("El bit 3 de 0x0f es correcto", !byte_nth_bit(0x0f,3));
  print_test("El bit 4 de 0x0f es correcto", byte_nth_bit(0x0f,4));
  print_test("El bit 5 de 0x0f es correcto", byte_nth_bit(0x0f,5));
  print_test("El bit 6 de 0x0f es correcto", byte_nth_bit(0x0f,6));
  print_test("El bit 7 de 0x0f es correcto", byte_nth_bit(0x0f,7));

  print_test("El bit 0 de 0xf0 es correcto", byte_nth_bit(0xf0,0));
  print_test("El bit 1 de 0xf0 es correcto", byte_nth_bit(0xf0,1));
  print_test("El bit 2 de 0xf0 es correcto", byte_nth_bit(0xf0,2));
  print_test("El bit 3 de 0xf0 es correcto", byte_nth_bit(0xf0,3));
  print_test("El bit 4 de 0xf0 es correcto", !byte_nth_bit(0xf0,4));
  print_test("El bit 5 de 0xf0 es correcto", !byte_nth_bit(0xf0,5));
  print_test("El bit 6 de 0xf0 es correcto", !byte_nth_bit(0xf0,6));
```

```
58    print_test("El bit 7 de 0xf0 es correcto", !byte_nth_bit(0xf0,7));
59  }
60
61  void test_byte_mask_nth_bit(){
62    printf("\n");
63    printf("--- INICIO DE PRUEBAS DE BYTE_MASK_NTH_BIT ---\n");
64
65    print_test("El bit 0 de 'a' es correcto", !(byte_mask_nth_bit('a',0)==128) );
66    print_test("El bit 1 de 'a' es correcto", byte_mask_nth_bit('a',1)==64);
67    print_test("El bit 2 de 'a' es correcto", byte_mask_nth_bit('a',2)==32);
68    print_test("El bit 3 de 'a' es correcto", !(byte_mask_nth_bit('a',3)==16) );
69    print_test("El bit 4 de 'a' es correcto", !(byte_mask_nth_bit('a',4)==8) );
70    print_test("El bit 5 de 'a' es correcto", !(byte_mask_nth_bit('a',5)==4) );
71    print_test("El bit 6 de 'a' es correcto", !(byte_mask_nth_bit('a',6)==2) );
72    print_test("El bit 7 de 'a' es correcto", byte_mask_nth_bit('a',7)==1);
73  }
74
75  void test_byte_extract_n_bits(){
76    printf("\n");
77    printf("--- INICIO DE PRUEBAS DE BYTE_EXTRACT_N_BITS ---\n");
78
79    print_test("Si agarro 'a' con un slice sigue siendo 'a'", byte_extract_n_bits
        ('a', 0, 7)=='a');
80    print_test("Si agarro 97 con un bit menos se conviente en 96",
        byte_extract_n_bits('a', 0, 6)==96);
81    print_test("Si agarro 97[1:6] se conviente en 96", byte_extract_n_bits('a',
        1, 6)==96);
82    print_test("Si agarro 97[2:6] se conviente en 32", byte_extract_n_bits('a',
        2, 6)==32);
83    print_test("Si agarro 1[7:7] obtengo 1", byte_extract_n_bits(1, 7, 7)==1);
84    print_test("Si agarro -128[0:0] obtengo -128", byte_extract_n_bits(-128, 0,
        0)==-128);
85    print_test("Si agarro r[2:7] obtengo 50", byte_extract_n_bits('r', 2, 7)==50)
        ;
86  }
87
88  void test_bytes_nth_bit(){
89    printf("\n");
90    printf("--- INICIO DE PRUEBAS DE BYTES_NTH_BIT ---\n");
91    char arreglo[]={'a','*',(char)0xff,(char)0x0f,(char)0xf0};
92    print_test("El bit 0 de 'a' es correcto", !bytes_nth_bit(arreglo,0));
93    print_test("El bit 1 de 'a' es correcto", bytes_nth_bit(arreglo,1));
94    print_test("El bit 2 de 'a' es correcto", bytes_nth_bit(arreglo,2));
95    print_test("El bit 3 de 'a' es correcto", !bytes_nth_bit(arreglo,3));
96    print_test("El bit 4 de 'a' es correcto", !bytes_nth_bit(arreglo,4));
97    print_test("El bit 5 de 'a' es correcto", !bytes_nth_bit(arreglo,5));
98    print_test("El bit 6 de 'a' es correcto", !bytes_nth_bit(arreglo,6));
99    print_test("El bit 7 de 'a' es correcto", bytes_nth_bit(arreglo,7));
100
101   print_test("El bit 0 de '*' es correcto", !bytes_nth_bit(arreglo,0+8));
102   print_test("El bit 1 de '*' es correcto", !bytes_nth_bit(arreglo,1+8));
103   print_test("El bit 2 de '*' es correcto", bytes_nth_bit(arreglo,2+8));
104   print_test("El bit 3 de '*' es correcto", !bytes_nth_bit(arreglo,3+8));
105   print_test("El bit 4 de '*' es correcto", bytes_nth_bit(arreglo,4+8));
106   print_test("El bit 5 de '*' es correcto", !bytes_nth_bit(arreglo,5+8));
107   print_test("El bit 6 de '*' es correcto", bytes_nth_bit(arreglo,6+8));
108   print_test("El bit 7 de '*' es correcto", !bytes_nth_bit(arreglo,7+8));
```

```
109
110   print_test("El bit 0 de 0xff es correcto", bytes_nth_bit(arreglo,0+16));
111   print_test("El bit 1 de 0xff es correcto", bytes_nth_bit(arreglo,1+16));
112   print_test("El bit 2 de 0xff es correcto", bytes_nth_bit(arreglo,2+16));
113   print_test("El bit 3 de 0xff es correcto", bytes_nth_bit(arreglo,3+16));
114   print_test("El bit 4 de 0xff es correcto", bytes_nth_bit(arreglo,4+16));
115   print_test("El bit 5 de 0xff es correcto", bytes_nth_bit(arreglo,5+16));
116   print_test("El bit 6 de 0xff es correcto", bytes_nth_bit(arreglo,6+16));
117   print_test("El bit 7 de 0xff es correcto", bytes_nth_bit(arreglo,7+16));
118
119   print_test("El bit 0 de 0x0f es correcto", !bytes_nth_bit(arreglo,0+24));
120   print_test("El bit 1 de 0x0f es correcto", !bytes_nth_bit(arreglo,1+24));
121   print_test("El bit 2 de 0x0f es correcto", !bytes_nth_bit(arreglo,2+24));
122   print_test("El bit 3 de 0x0f es correcto", !bytes_nth_bit(arreglo,3+24));
123   print_test("El bit 4 de 0x0f es correcto", bytes_nth_bit(arreglo,4+24));
124   print_test("El bit 5 de 0x0f es correcto", bytes_nth_bit(arreglo,5+24));
125   print_test("El bit 6 de 0x0f es correcto", bytes_nth_bit(arreglo,6+24));
126   print_test("El bit 7 de 0x0f es correcto", bytes_nth_bit(arreglo,7+24));
127 }
128
129 void test_bytes_extract_n_bits(){
130   printf("\n");
131   printf("--- INICIO DE PRUEBAS DE BYTES_EXTRACT_N_BITS ---\n");
132   char arreglo[] = {'a',1,-128};
133   print_test("Si agarro 'a' con un slice sigue siendo 'a'", *
        bytes_extract_n_bits(arreglo, 0, 7)=='a');
134   print_test("Si agarro 97 con un bit menos se conviente en 96", *
        bytes_extract_n_bits(arreglo, 0, 6)==96);
135   print_test("Si agarro 97[1:6] se conviente en 96", *bytes_extract_n_bits(
        arreglo, 1, 6)==96);
136   print_test("Si agarro 97[2:6] se conviente en 32", *bytes_extract_n_bits(
        arreglo, 2, 6)==32);
137   print_test("Si agarro 1[7:7] obtengo 1", *bytes_extract_n_bits(arreglo, 7+8,
        7+8)==1);
138   print_test("Si agarro -128[0:0] obtengo -128", *bytes_extract_n_bits(arreglo,
         16, 16)==-128);
139
140   print_test("Si agarro arreglo[0:23][0] obtengo arreglo[0]",
        bytes_extract_n_bits(arreglo, 0, 23)[0]==arreglo[0]);
141   print_test("Si agarro arreglo[0:23][1] obtengo arreglo[1]",
        bytes_extract_n_bits(arreglo, 0, 23)[1]==arreglo[1]);
142   print_test("Si agarro arreglo[0:23][2] obtengo arreglo[2]",
        bytes_extract_n_bits(arreglo, 0, 23)[2]==arreglo[2]);
143
144   print_test("Si agarro arreglo[8:23][1] obtengo arreglo[2]",
        bytes_extract_n_bits(arreglo, 8, 23)[1]==arreglo[2]);
145   print_test("Si agarro arreglo[8:15][0] obtengo arreglo[1]",
        bytes_extract_n_bits(arreglo, 8, 15)[0]==arreglo[1]);
146   print_test("Si agarro arreglo[16:16][0] obtengo arreglo[2]",
        bytes_extract_n_bits(arreglo, 16, 16)[0]==arreglo[2]);
147
148   char arreglo2[] = {'*' , 'f', 'r'};
149   print_test("Si agarro arreglo[18:23][0] obtengo 50", bytes_extract_n_bits(
        arreglo2, 18, 23)[0]==50);
150 }
151
152 int main(){
```

```
153   test_byte_nth_bit();
154   test_byte_mask_nth_bit();
155   test_byte_extract_n_bits();
156   test_bytes_nth_bit();
157   test_bytes_extract_n_bits();
158   return EXIT_SUCCESS;
159 }
```

base64.h

```
1  #include <stdlib.h>
2
3  char* base64_transform_3bytes(char* bytes);
4
5  int base64_encode_bytes(char* bytes, int tam, char** dest);
6
7  char decode_byte(char byte);
8
9  int base64_decode_4bytes(char* bytes, char** dest);
10
11 int base64_decode_bytes(char* bytes, int tam, char** dest);
```

base64.c

```c
#include "bit_library.h"
#include "base64.h"
#include <string.h>
#include <stdio.h>

static const char base64_table[] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M' ,'N', 'O', 'P',
                                    'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'a', 'b', 'c', 'd', 'e', 'f',
                                    'g', 'h', 'i', 'j', 'k', 'l', 'm' ,'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',
                                    'w', 'x', 'y', 'z','0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '+', '/'};

char decode_byte(char byte){
  if(byte>='A' && byte<='Z') return (byte-'A');
  if(byte>='a' && byte<='z') return (byte-'a')+26;
  if(byte>='0' && byte<='9') return (byte-'0')+52;
  if(byte=='+') return 62;
  if(byte=='/') return 63;
  return -1;
}

char* base64_transform_3bytes(char* bytes){
  char* base64 = (char*)malloc(4);
  if(!base64) return NULL;
  for(int i=0;i<4;i++){
    char* extracted = bytes_extract_n_bits(bytes, i*6, i*6+6-1);
    if(!extracted){
      free(base64);
      return NULL;
    }
    if(i==0){
      base64[i] = (unsigned char)extracted[0] >> 2;
    } else if(i==1){
      base64[i] = (unsigned char)extracted[0] << 4 | (unsigned char)extracted[1] >> 4;
    } else if(i==2){
      base64[i] = (unsigned char)extracted[0] << 2 | (unsigned char)extracted[1] >> 6;
    } else {
      base64[i] = (unsigned char)extracted[0];
    }
    free(extracted);
    base64[i] = base64_table[base64[i]];
  }
  return base64;
}

int base64_encode_bytes(char* bytes, int tam, char** dest){
  int offset = tam%3>0 ? abs(tam%3-3) : 0;
  int bytes_to_transform = tam + offset;

  char* copy = (char*) calloc(bytes_to_transform, 1);
  if(!copy) return -1;
  memcpy(copy, bytes, tam);
```

```
51    bytes = copy;
52    tam = bytes_to_transform;
53
54    int total_bits = (tam*8);
55    int result_bytes = total_bits/6;
56    char* result = (char*) malloc(result_bytes);
57    if(!result){
58      return -1;
59      free(copy);
60    }
61    for(int i=0; i < bytes_to_transform/3;i++){
62      char* encoding = base64_transform_3bytes(copy+i*3);
63      result[i*4] = encoding[0];
64      result[i*4+1] = encoding[1];
65      result[i*4+2] = encoding[2];
66      result[i*4+3] = encoding[3];
67      free(encoding);
68    }
69    for(int i=result_bytes-offset;i<result_bytes;i++){
70      result[i]='=';
71    }
72    free(copy);
73    *dest = result;
74    return result_bytes;
75  }
76
77  int base64_decode_4bytes(char* bytes, char** dest){
78    int iguales = (bytes[2]=='=' ? 1:0) + (bytes[3]=='=' ? 1:0);
79    char* aux = (char*)calloc(3,1);
80    if(!aux) return -1;
81    char* copy = (char*)malloc(4);
82    memcpy(copy, bytes, 4);
83    for(int i=0;i<4;i++){
84      if(decode_byte(copy[i])==-1 && copy[i]!='='){
85        free(copy);
86        free(aux);
87        return -1;
88      }
89      copy[i] = decode_byte(copy[i]);
90    }
91    for(int i=0;i<4;i++){
92      char* extracted = bytes_extract_n_bits(copy, i*8, i*8+8-1);
93      if(!extracted){
94        free(aux);
95        free(copy);
96        return -1;
97      }
98      if(i==0){
99        aux[0] |= byte_extract_n_bits(extracted[0], 2, 7) << 2;
100     } else if(i==1){
101       aux[0] |= byte_extract_n_bits(extracted[0], 2, 4) >> 4;
102       aux[1] |= byte_extract_n_bits(extracted[0], 4, 7) << 4;
103     } else if(i==2){
104       aux[1] |= byte_extract_n_bits(extracted[0], 2, 5) >> 2;
105       aux[2] |= byte_extract_n_bits(extracted[0], 5, 7) << 6;
106     } else {
107       aux[2] |= byte_extract_n_bits(extracted[0], 2, 7);
```

```
108       }
109       free(extracted);
110     }
111     char* resized = (char*) malloc(3-iguales);
112     if(!resized){
113       free(aux);
114       free(copy);
115       return -1;
116     }
117     memcpy(resized, aux, 3-iguales);
118     *dest = resized;
119     free(copy);
120     free(aux);
121     return 3-iguales;
122 }
123
124 int base64_decode_bytes(char* bytes, int tam, char** dest){
125     char* result = NULL;
126     if(tam%4!=0) return -1;
127     char* decoded;
128     int longitud;
129     int longitud_total = 0;
130     for(int i=0;i<tam/4;i++){
131       longitud = base64_decode_4bytes(bytes+i*4, &decoded);
132       if(longitud<0){
133         free(result);
134         return -1;
135       }
136       if(!result) result = (char*) malloc(longitud);
137       else {
138         char* aux = (char*)realloc(result, longitud_total+longitud);
139         if(!aux){
140           free(result);
141           free(decoded);
142           return -1;
143         }
144         result = aux;
145       }
146       memcpy(result+longitud_total, decoded, longitud);
147       longitud_total += longitud;
148       free(decoded);
149     }
150     *dest = result;
151     return longitud_total;
152 }
```

test_base64.c

```c
#include "base64.h"
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>

void print_test(const char* msg, bool cond){
  if(cond) printf("%s ... %s", msg, "OK");
  else printf("%s ... %s", msg, "ERROR");
  printf("\n");
}

bool byte_compare(const char* cadena1, const char* cadena2, size_t tam){
  char comparer = 0x00;
  for(size_t i=0;i<tam;i++){
      comparer |= cadena1[i] ^ cadena2[i];
  }
  return comparer==0;
}

void test_encode_3bytes(){
  printf("\n");
  printf("--- INICIO DE PRUEBAS ENCODE 3 BYTES ---\n");

  char prueba1[] = {'a','s','d'};
  char* encoding = base64_transform_3bytes(prueba1);
  print_test("Si encodeo 'asd' recibo 'YXNk'", byte_compare(encoding, "YXNk",
    4));
  free(encoding);

  char prueba2[] = {'$','%','$'};
  encoding = base64_transform_3bytes(prueba2);
  print_test("Si encodeo '$%$' recibo 'JCUk'", byte_compare(encoding, "JCUk",
    4));
  free(encoding);

  char prueba3[] = {'*','f','r'};
  encoding = base64_transform_3bytes(prueba3);
  print_test("Si encodeo '*fr' recibo 'KmZy'", byte_compare(encoding, "KmZy",
    4));
  free(encoding);

  char prueba4[] = {0x2E, 0xF9, 0x49};
  const char* salida_esperada4="LvlJ";
  encoding = base64_transform_3bytes(prueba4);
  print_test("Caso de prueba 4", byte_compare(encoding, salida_esperada4, 4));
  free(encoding);
}

void test_encode_base64(){
  printf("\n");
  printf("--- INICIO DE PRUEBAS ENCODE STRING ---\n");
  char* encoding;

  char prueba1[] = {'a','s','d'};
  int longitud = base64_encode_bytes(prueba1, 3, &encoding);
```

```
54    print_test("Si encodeo 'asd' recibo 'YXNk'", byte_compare(encoding, "YXNk",
        4));
55    free(encoding);
56
57    char prueba2[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g'};
58    longitud = base64_encode_bytes(prueba2, 7, &encoding);
59    print_test("Si encodeo 'abcdefg' recibo 'YWJjZGVmZw=='", byte_compare(
        encoding, "YWJjZGVmZw==", 12));
60    free(encoding);
61
62    char prueba3[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'};
63    longitud = base64_encode_bytes(prueba3, 8, &encoding);
64    print_test("Si encodeo 'abcdefgh' recibo 'YWJjZGVmZ2g='", byte_compare(
        encoding, "YWJjZGVmZ2g=", 12));
65    free(encoding);
66
67    const char* prueba4 = "Man is distinguished, not only by his reason, but by
        this singular passion from other animals, which is a lust of the mind, that
        by a perseverance of delight in the continued and indefatigable generation
        of knowledge, exceeds the short vehemence of any carnal pleasure.";
68    longitud = base64_encode_bytes((char*) prueba4, strlen(prueba4), &encoding);
69    const char* salida_esperada4 = "
        TWFuIGlzIGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmx5IGJ5IGhpcyByZWFzb24sIGJ1dCBieSB0aGlzIHNpbmd1bGFyIHBhc3
        =";
70    print_test("Encodeo un texto largo (wikipedia)", byte_compare(encoding,
        salida_esperada4, longitud));
71    print_test("La longitud devuelta es correcta", longitud==strlen(
        salida_esperada4));
72    free(encoding);
73  }
74
75  void test_decode_byte(){
76    printf("\n");
77    printf("--- INICIO DE PRUEBAS DECODE BYTE ---\n");
78
79    print_test("Decode 'A' es 0", decode_byte('A')==0);
80    print_test("Decode 'a' es 26", decode_byte('a')==26);
81    print_test("Decode '0' es 52", decode_byte('0')==52);
82    print_test("Decode '+' es 62", decode_byte('+')==62);
83    print_test("Decode '/' es 63", decode_byte('/')==63);
84  }
85
86  void test_decode_4bytes(){
87    printf("\n");
88    printf("--- INICIO DE PRUEBAS DECODE 4 BYTES ---\n");
89
90    char prueba1[] = {'Y','X','N','k'};
91    char* decode;
92    int longitud = base64_decode_4bytes(prueba1, &decode);
93    print_test("Si desencodeo 'YXNk' recibo 'asd'", byte_compare(decode, "asd",
        longitud));
94    free(decode);
95
96    char prueba2[] = {'J', 'C', 'U', 'k'};
97    longitud = base64_decode_4bytes(prueba2, &decode);
98    print_test("Si desencodeo 'JCUk' recibo '$%$'", byte_compare(decode, "$%$",
        longitud));
```

```
99    free(decode);
100
101   char prueba3[] = {'K', 'm', 'Z', 'y'};
102   longitud = base64_decode_4bytes(prueba3, &decode);
103   print_test("Si desencodeo 'KmZy' recibo '*fr'", byte_compare(decode, "*fr",
        longitud));
104   free(decode);
105
106   char prueba4[] = {'J', 'j', 'g', '='};
107   longitud = base64_decode_4bytes(prueba4, &decode);
108   print_test("Si desencodeo 'Jjg=' recibo '&8'", byte_compare(decode, "&8",
        longitud));
109   free(decode);
110
111   char prueba5[] = {'J', 'j', '=', '='};
112   longitud = base64_decode_4bytes(prueba5, &decode);
113   print_test("Si desencodeo 'Jj==' recibo '&'", byte_compare(decode, "&",
        longitud));
114   free(decode);
115
116   char prueba6[] = {'@', 'j', '=', '='};
117   longitud = base64_decode_4bytes(prueba6, &decode);
118   print_test("Si desencodeo '@j==' falla", longitud==-1);
119 }
120
121 void test_decode_bytes(){
122   printf("\n");
123   printf("--- INICIO DE PRUEBAS DECODE BYTES ---\n");
124   char* decode;
125
126   char prueba1[] = {'Y', 'X', 'N', 'k'};
127   int longitud = base64_decode_bytes(prueba1, 4, &decode);
128   print_test("Si desencodeo 'YXNk' recibo 'asd'", byte_compare(decode, "asd",
        3));
129   free(decode);
130
131   char prueba2[] = {'Y','W','J','j','Z','G','V','m','Z','w','=','='};
132   longitud = base64_decode_bytes(prueba2, 12, &decode);
133   print_test("Si desencodeo 'YWJjZGVmZw==' recibo 'abcdefg'", byte_compare(
        decode, "abcdefg", longitud));
134   free(decode);
135
136   char prueba3[] = {'Y', 'W', 'J', 'j', 'Z', 'G', 'V', 'm', 'Z', '2', 'g', '='
        };
137   longitud = base64_decode_bytes(prueba3, 12, &decode);
138   print_test("Si desencodeo 'YWJjZGVmZ2g=' recibo 'abcdefgh'", byte_compare(
        decode, "abcdefgh", longitud));
139   free(decode);
140
141   const char* prueba4 = "
        TWFuIGlzIGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmx5IGJ5IGhpcyByZWFzb24sIGJ1dCBieSB0aGlzIHNpbmd1bGFyIHBhc3
        =";
142   const char* salida_esperada = "Man is distinguished, not only by his reason,
        but by this singular passion from other animals, which is a lust of the mind
        , that by a perseverance of delight in the continued and indefatigable
        generation of knowledge, exceeds the short vehemence of any carnal pleasure.
        ";
```

```
143   longitud = base64_decode_bytes((char*) prueba4, strlen(prueba4), &decode);
144   print_test("Desencodeo un texto largo (wikipedia)", byte_compare(decode,
        salida_esperada, longitud));
145   print_test("La longitud devuelta es correcta", longitud==strlen(
        salida_esperada));
146   free(decode);
147
148   char prueba5[] = {'Y', 'W', 'J', 'j', 'Z', 'G', '#', 'm', 'Z', '2', 'g', '='
        };
149   longitud = base64_decode_bytes(prueba5, 12, &decode);
150   print_test("Si desencodeo 'YWJjZG#mZ2g=' falla", longitud=-1);
151
152   char prueba6[] = {'Y', 'W', 'J', 'j', 'Z', 'G', 'V', 'm', 'Z', '2', 'g'};
153   longitud = base64_decode_bytes(prueba5, 11, &decode);
154   print_test("Si desencodeo 'YWJjZG#mZ2g' falla", longitud=-1);
155 }
156
157 int main(){
158   test_encode_3bytes();
159   test_encode_base64();
160   test_decode_byte();
161   test_decode_4bytes();
162   test_decode_bytes();
163   return EXIT_SUCCESS;
164 }
```

main.c

```c
#define _POSIX_C_SOURCE 200809L
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <stdbool.h>
#include "base64.h"

#define BYTES_A_LEER 120

#define AYUDA "-h"
#define VERSION "-V"
#define version_imprimible "1.0"
#define ERROR_PARAMETROS "Invalid parameters. Use help menu and try again."
#define ERROR_CANTIDAD_PARAMETROS "Invalid number of parameters."

#define INPUT_FILE "-i"
#define OUTPUT_FILE "-o"
#define ACTION "-a"
#define ERROR_ARCHIVO "Invalid file. Please use a file that exist or check the path."
#define ERROR_NO_INPUT_FILE "Missing input file. Try again."
#define ERROR_NO_OUTPUT_FILE "Missing output file. Try again."


#define ENCODE "encode"
#define DECODE "decode"
#define ERROR_ACCION "Invalid action. Please use an allowed action. See the help menu."
#define ERROR_NO_ACCION "Please, specify the action. Default action is encoding."
#define ERROR_PERFORMING_ACTION "An error ocurred while trying to perform your request."

void imprimir_ayuda(){
  printf("\
  Usage:\
  tp0 -h\n\
  tp0 -V\n\
  tp0 [options]\n\
  Options:\n\
  -V, --version\n\
  Print version and quit.\
  -h, --help\n\
  Print this information.\
  -i, --input\n\
  Location of the input file.\n\
  -o, --output\n\
  Location of the output file.\n\
  -a, --action\n\
  Program action: encode (default) or decode.\n\
  Examples:\n\
  tp0 -a encode -i ~/input -o ~/output\n\
  tp0 -a decode\n\
  ");
}

```

```
53
54 int encode_file(FILE* input_file, FILE* output_file){
55   bool stop = false;
56   int longitud;
57   while (!stop){
58     int byte_leido;
59     char* encoding;
60     char* linea = (char*)malloc(sizeof(char)*BYTES_A_LEER);
61     if(!linea) return -1;
62     int cantidad_bytes_leidos = 0;
63     while (cantidad_bytes_leidos < BYTES_A_LEER && (byte_leido = fgetc(
       input_file)) != EOF)  {
64       linea[cantidad_bytes_leidos] = (char)byte_leido;
65       cantidad_bytes_leidos++;
66     }
67     if(byte_leido == EOF) stop = true;
68     if(cantidad_bytes_leidos != 0){
69       longitud = base64_encode_bytes(linea, cantidad_bytes_leidos, &encoding);
70       if(longitud<0){
71         free(linea);
72         return -1;
73       }
74       fwrite((void*)encoding, 1, longitud, output_file);
75       free(encoding);
76     }
77     free(linea);
78   }
79
80   return EXIT_SUCCESS;
81 }
82
83 int decode_file(FILE* input_file, FILE* output_file){
84   bool stop = false;
85   int longitud;
86   while(!stop){
87     char* linea = malloc(sizeof(char)*BYTES_A_LEER);
88     if(!linea) return -1;
89     int byte_leido;
90     char* decoding;
91     int i;
92     int cantidad_bytes_leidos = 0;
93     while (cantidad_bytes_leidos < BYTES_A_LEER && (byte_leido = fgetc(
       input_file)) != EOF) {
94       linea[cantidad_bytes_leidos] = (char)byte_leido;
95       cantidad_bytes_leidos++;
96     }
97     if(byte_leido == EOF) stop = true;
98     if(cantidad_bytes_leidos != 0){
99       longitud = base64_decode_bytes(linea, cantidad_bytes_leidos, &decoding);
100      if(longitud<0){
101        return -1;
102      }
103      fwrite((void*)decoding, 1, longitud, output_file);
104      free(decoding);
105    }
106    free(linea);
107  }
```

```
108   return EXIT_SUCCESS;
109 }
110
111 int main(int argc, char* argv[]){
112   int i = 1;
113   int result=0;
114
115   //default encode entrada estandar.
116   if (argc == 1){
117     int result = encode_file(stdin, stdout);
118     if(result!=0) fprintf(stderr, "%s\n", ERROR_PERFORMING_ACTION);
119     return result;
120   }
121
122   //ayuda o version.
123   if (argc == 2){
124     if (strcmp(argv[i], AYUDA) == 0)
125       imprimir_ayuda();
126     else if (strcmp(argv[i], VERSION) == 0)
127       printf("%s\n", version_imprimible);
128     else if (strcmp(argv[i], ACTION) == 0)
129       fprintf(stderr, "%s\n", ERROR_NO_ACCION);
130     else
131       fprintf(stderr, "%s\n", ERROR_PARAMETROS);
132     return -1;
133   }
134
135   //encode o decode.
136   else if (argc > 2 && argc < 8){
137     if(strcmp(argv[i], ACTION) == 0){
138       i++;
139       bool encode;
140       if (strcmp(argv[i], ENCODE) == 0) encode = true;
141       else if (strcmp(argv[i], DECODE) == 0) encode = false;
142       else{
143         fprintf(stderr, "%s\n", ERROR_ACCION);
144         return -1;
145       }
146       i++;
147       //caso ./main -a encode(decode) sin archivos.
148       if (!argv[i]){
149         if (encode)
150           result = encode_file(stdin, stdout);
151         else
152           result = decode_file(stdin, stdout);
153         if(result!=0) fprintf(stderr, "%s\n", ERROR_PERFORMING_ACTION);
154         return result;
155       }
156       //./main -a encode(decode) -i inputfile
157       if (strcmp(argv[i], INPUT_FILE) == 0){
158         i++;
159         if(!argv[i]){
160           fprintf(stderr, "%s\n", ERROR_NO_INPUT_FILE);
161           return -1;
162         }
163         FILE* input_file = fopen(argv[i], "r");
164         if(!input_file){
```

```
165             fprintf(stderr, "%s\n", ERROR_ARCHIVO);
166             return -1;
167           }
168           i++;
169           if(!argv[i]){
170             fprintf(stderr, "%s\n", ERROR_NO_OUTPUT_FILE);
171             fclose(input_file);
172             return -1;
173           }
174           // ./main -a encode(decode) -i inputfile -o outputfile
175           if(strcmp(argv[i], OUTPUT_FILE) == 0){
176             i++;
177             if(!argv[i]){
178               fprintf(stderr, "%s\n", ERROR_NO_OUTPUT_FILE);
179               fclose(input_file);
180               return -1;
181             }
182             FILE* output_file = fopen(argv[i], "w");
183             if(encode){
184               result = encode_file(input_file, output_file);
185             }
186             else{
187               result = decode_file(input_file, output_file);
188             }
189             if(result!=0) fprintf(stderr, "%s\n", ERROR_PERFORMING_ACTION);
190             return result;
191           }
192           else{
193             fprintf(stderr, "%s\n", ERROR_PARAMETROS);
194             fclose(input_file);
195             return -1;
196           }
197         }
198         else{
199           fprintf(stderr, "%s\n", ERROR_PARAMETROS);
200           return -1;
201         }
202       }
203     }
204     else{
205       fprintf(stderr, "%s\n", ERROR_CANTIDAD_PARAMETROS);
206       return -1;
207     }
208     return EXIT_SUCCESS;
209 }
```

Código MIPS32

Al realizar -objdump -d main obtenemos que el código assembly generado es:

```
00400914 <_init>:
  400914:       3c1c0fc0        lui     gp,0xfc0
  400918:       279c771c        addiu   gp,gp,30492
  40091c:       0399e021        addu    gp,gp,t9
```

```
  400920:       27bdffe0        addiu   sp,sp,-32
  400924:       afbc0010        sw      gp,16(sp)
  400928:       afbf001c        sw      ra,28(sp)
  40092c:       afbc0018        sw      gp,24(sp)
  400930:       8f998018        lw      t9,-32744(gp)
  400934:       00000000        nop
  400938:       2739095c        addiu   t9,t9,2396
  40093c:       00000000        nop
  400940:       0320f809        jalr    t9
  400944:       00000000        nop
  400948:       8fbc0010        lw      gp,16(sp)
  40094c:       8fbf001c        lw      ra,28(sp)
  400950:       00000000        nop
  400954:       03e00008        jr      ra
  400958:       27bd0020        addiu   sp,sp,32

0040095c <init_fallthru>:
  40095c:       3c1c0fc0        lui     gp,0xfc0
  400960:       279c76d4        addiu   gp,gp,30420
  400964:       0399e021        addu    gp,gp,t9
  400968:       27bdffe0        addiu   sp,sp,-32
  40096c:       afbc0010        sw      gp,16(sp)
  400970:       afbf001c        sw      ra,28(sp)
  400974:       04110001        bal     40097c <init_fallthru+0x20>
  400978:       00000000        nop
  40097c:       3c1c0fc0        lui     gp,0xfc0
  400980:       279c76b4        addiu   gp,gp,30388
  400984:       039fe021        addu    gp,gp,ra
  400988:       8f998018        lw      t9,-32744(gp)
  40098c:       00000000        nop
  400990:       27390ddc        addiu   t9,t9,3548
  400994:       0320f809        jalr    t9
  400998:       00000000        nop
  40099c:       8fbc0010        lw      gp,16(sp)
  4009a0:       00000000        nop
  4009a4:       04110001        bal     4009ac <init_fallthru+0x50>
  4009a8:       00000000        nop
  4009ac:       3c1c0fc0        lui     gp,0xfc0
  4009b0:       279c7684        addiu   gp,gp,30340
  4009b4:       039fe021        addu    gp,gp,ra
  4009b8:       8f998018        lw      t9,-32744(gp)
  4009bc:       00000000        nop
  4009c0:       273932e0        addiu   t9,t9,13024
  4009c4:       0320f809        jalr    t9
  4009c8:       00000000        nop
  4009cc:       8fbc0010        lw      gp,16(sp)
  4009d0:       00000000        nop
```

```
  4009d4:       8fbf001c        lw      ra,28(sp)
  4009d8:       00000000        nop
  4009dc:       03e00008        jr      ra
  4009e0:       27bd0020        addiu   sp,sp,32
Disassembly of section .text:

004009f0 <__start>:
  4009f0:       3c1c0fc0        lui     gp,0xfc0
  4009f4:       279c7640        addiu   gp,gp,30272
  4009f8:       0399e021        addu    gp,gp,t9
  4009fc:       27bdffc0        addiu   sp,sp,-64
  400a00:       afbc0010        sw      gp,16(sp)
  400a04:       afb60030        sw      s6,48(sp)
  400a08:       afb5002c        sw      s5,44(sp)
  400a0c:       afb40028        sw      s4,40(sp)
  400a10:       afb1001c        sw      s1,28(sp)
  400a14:       00a0a821        move    s5,a1
  400a18:       afbf0038        sw      ra,56(sp)
  400a1c:       afbc0034        sw      gp,52(sp)
  400a20:       afb30024        sw      s3,36(sp)
  400a24:       afb20020        sw      s2,32(sp)
  400a28:       afb00018        sw      s0,24(sp)
  400a2c:       00808821        move    s1,a0
  400a30:       00c0b021        move    s6,a2
  400a34:       00e0a021        move    s4,a3
  400a38:       14800002        bnez    a0,400a44 <__start+0x54>
  400a3c:       00802821        move    a1,a0
  400a40:       27a50040        addiu   a1,sp,64
  400a44:       8cb20000        lw      s2,0(a1)
  400a48:       8cb00004        lw      s0,4(a1)
  400a4c:       00121080        sll     v0,s2,0x2
  400a50:       00a21021        addu    v0,a1,v0
  400a54:       24420008        addiu   v0,v0,8
  400a58:       8f81808c        lw      at,-32628(gp)
  400a5c:       00000000        nop
  400a60:       ac220000        sw      v0,0(at)
  400a64:       12000012        beqz    s0,400ab0 <__start+0xc0>
  400a68:       24b30004        addiu   s3,a1,4
  400a6c:       02002021        move    a0,s0
  400a70:       2405002f        li      a1,47
  400a74:       8f998018        lw      t9,-32744(gp)
  400a78:       00000000        nop
  400a7c:       27390b90        addiu   t9,t9,2960
  400a80:       00000000        nop
  400a84:       0320f809        jalr    t9
  400a88:       00000000        nop
  400a8c:       8fbc0010        lw      gp,16(sp)
```

```
400a90:       00000000        nop
400a94:       8f81807c        lw      at,-32644(gp)
400a98:       00000000        nop
400a9c:       14400037        bnez    v0,400b7c <__start+0x18c>
400aa0:       ac220000        sw      v0,0(at)
400aa4:       8f81807c        lw      at,-32644(gp)
400aa8:       00000000        nop
400aac:       ac300000        sw      s0,0(at)
400ab0:       12200015        beqz    s1,400b08 <__start+0x118>
400ab4:       00000000        nop
400ab8:       12800004        beqz    s4,400acc <__start+0xdc>
400abc:       00000000        nop
400ac0:       8f818044        lw      at,-32700(gp)
400ac4:       00000000        nop
400ac8:       ac340000        sw      s4,0(at)
400acc:       8f8280a8        lw      v0,-32600(gp)
400ad0:       00000000        nop
400ad4:       1040000c        beqz    v0,400b08 <__start+0x118>
400ad8:       00000000        nop
400adc:       12a0000a        beqz    s5,400b08 <__start+0x118>
400ae0:       00000000        nop
400ae4:       12c00008        beqz    s6,400b08 <__start+0x118>
400ae8:       02a02021        move    a0,s5
400aec:       02c02821        move    a1,s6
400af0:       8f99803c        lw      t9,-32708(gp)
400af4:       00000000        nop
400af8:       0320f809        jalr    t9
400afc:       00000000        nop
400b00:       8fbc0010        lw      gp,16(sp)
400b04:       00000000        nop
400b08:       8f84805c        lw      a0,-32676(gp)
400b0c:       8f998054        lw      t9,-32684(gp)
400b10:       00000000        nop
400b14:       0320f809        jalr    t9
400b18:       00000000        nop
400b1c:       8fbc0010        lw      gp,16(sp)
400b20:       00000000        nop
400b24:       8f998094        lw      t9,-32620(gp)
400b28:       00000000        nop
400b2c:       0320f809        jalr    t9
400b30:       00000000        nop
400b34:       8fbc0010        lw      gp,16(sp)
400b38:       00000000        nop
400b3c:       8f86808c        lw      a2,-32628(gp)
400b40:       00000000        nop
400b44:       8cc60000        lw      a2,0(a2)
400b48:       02402021        move    a0,s2
```

```
  400b4c:       02602821        move    a1,s3
  400b50:       8f998068        lw      t9,-32664(gp)
  400b54:       00000000        nop
  400b58:       0320f809        jalr    t9
  400b5c:       00000000        nop
  400b60:       8fbc0010        lw      gp,16(sp)
  400b64:       00402021        move    a0,v0
  400b68:       8f998040        lw      t9,-32704(gp)
  400b6c:       00000000        nop
  400b70:       0320f809        jalr    t9
  400b74:       00000000        nop
  400b78:       8fbc0010        lw      gp,16(sp)
  400b7c:       24420001        addiu   v0,v0,1
  400b80:       8f81807c        lw      at,-32644(gp)
  400b84:       00000000        nop
  400b88:       1000ffc9        b       400ab0 <__start+0xc0>
  400b8c:       ac220000        sw      v0,0(at)
  400b90:       3c1c0fc0        lui     gp,0xfc0
  400b94:       279c74a0        addiu   gp,gp,29856
  400b98:       0399e021        addu    gp,gp,t9
  400b9c:       00001021        move    v0,zero
  400ba0:       80830000        lb      v1,0(a0)
  400ba4:       00000000        nop
  400ba8:       10650005        beq     v1,a1,400bc0 <__start+0x1d0>
  400bac:       00000000        nop
  400bb0:       1460fffb        bnez    v1,400ba0 <__start+0x1b0>
  400bb4:       24840001        addiu   a0,a0,1
  400bb8:       03e00008        jr      ra
  400bbc:       00000000        nop
  400bc0:       1000fffb        b       400bb0 <__start+0x1c0>
  400bc4:       00801021        move    v0,a0

00400bc8 <_rtld_setup>:
  400bc8:       3c1c0fc0        lui     gp,0xfc0
  400bcc:       279c7468        addiu   gp,gp,29800
  400bd0:       0399e021        addu    gp,gp,t9
  400bd4:       27bdffd0        addiu   sp,sp,-48
  400bd8:       afbc0018        sw      gp,24(sp)
  400bdc:       afb00020        sw      s0,32(sp)
  400be0:       3c03d550        lui     v1,0xd550
  400be4:       00a08021        move    s0,a1
  400be8:       afb10024        sw      s1,36(sp)
  400bec:       afbf002c        sw      ra,44(sp)
  400bf0:       00808821        move    s1,a0
  400bf4:       afbc0028        sw      gp,40(sp)
  400bf8:       8f878018        lw      a3,-32744(gp)
  400bfc:       00000000        nop
```

```
400c00:       24e73524        addiu   a3,a3,13604
400c04:       24060002        li      a2,2
400c08:       24050000        li      a1,0
400c0c:       24040004        li      a0,4
400c10:       12000005        beqz    s0,400c28 <_rtld_setup+0x60>
400c14:       3463b87a        ori     v1,v1,0xb87a
400c18:       8e020000        lw      v0,0(s0)
400c1c:       00000000        nop
400c20:       10430010        beq     v0,v1,400c64 <_rtld_setup+0x9c>
400c24:       00000000        nop
400c28:       24020021        li      v0,33
400c2c:       afa20010        sw      v0,16(sp)
400c30:       8f9980c4        lw      t9,-32572(gp)
400c34:       00000000        nop
400c38:       0320f809        jalr    t9
400c3c:       00000000        nop
400c40:       8fbc0018        lw      gp,24(sp)
400c44:       24050000        li      a1,0
400c48:       24040001        li      a0,1
400c4c:       24060001        li      a2,1
400c50:       8f9980c4        lw      t9,-32572(gp)
400c54:       00000000        nop
400c58:       0320f809        jalr    t9
400c5c:       00000000        nop
400c60:       8fbc0018        lw      gp,24(sp)
400c64:       8e030004        lw      v1,4(s0)
400c68:       24020001        li      v0,1
400c6c:       24050000        li      a1,0
400c70:       24040004        li      a0,4
400c74:       8f878018        lw      a3,-32744(gp)
400c78:       00000000        nop
400c7c:       24e73548        addiu   a3,a3,13640
400c80:       10620010        beq     v1,v0,400cc4 <_rtld_setup+0xfc>
400c84:       24060002        li      a2,2
400c88:       24020020        li      v0,32
400c8c:       afa20010        sw      v0,16(sp)
400c90:       8f9980c4        lw      t9,-32572(gp)
400c94:       00000000        nop
400c98:       0320f809        jalr    t9
400c9c:       00000000        nop
400ca0:       8fbc0018        lw      gp,24(sp)
400ca4:       24050000        li      a1,0
400ca8:       24040001        li      a0,1
400cac:       24060001        li      a2,1
400cb0:       8f9980c4        lw      t9,-32572(gp)
400cb4:       00000000        nop
400cb8:       0320f809        jalr    t9
```

```
  400cbc:       00000000        nop
  400cc0:       8fbc0018        lw      gp,24(sp)
  400cc4:       02202021        move    a0,s1
  400cc8:       8f998054        lw      t9,-32684(gp)
  400ccc:       00000000        nop
  400cd0:       0320f809        jalr    t9
  400cd4:       00000000        nop
  400cd8:       8fbc0018        lw      gp,24(sp)
  400cdc:       8fbf002c        lw      ra,44(sp)
  400ce0:       8fb10024        lw      s1,36(sp)
  400ce4:       8fb00020        lw      s0,32(sp)
  400ce8:       03e00008        jr      ra
  400cec:       27bd0030        addiu   sp,sp,48

00400cf0 <__do_global_dtors_aux>:
  400cf0:       3c1c0fc0        lui     gp,0xfc0
  400cf4:       279c7340        addiu   gp,gp,29504
  400cf8:       0399e021        addu    gp,gp,t9
  400cfc:       27bdffe0        addiu   sp,sp,-32
  400d00:       afbc0010        sw      gp,16(sp)
  400d04:       8f82801c        lw      v0,-32740(gp)
  400d08:       00000000        nop
  400d0c:       24420110        addiu   v0,v0,272
  400d10:       90420000        lbu     v0,0(v0)
  400d14:       afbf001c        sw      ra,28(sp)
  400d18:       1440002c        bnez    v0,400dcc <__do_global_dtors_aux+0xdc>
  400d1c:       afbc0018        sw      gp,24(sp)
  400d20:       8f82801c        lw      v0,-32740(gp)
  400d24:       00000000        nop
  400d28:       24420014        addiu   v0,v0,20
  400d2c:       8c420000        lw      v0,0(v0)
  400d30:       00000000        nop
  400d34:       8c430000        lw      v1,0(v0)
  400d38:       00000000        nop
  400d3c:       10600013        beqz    v1,400d8c <__do_global_dtors_aux+0x9c>
  400d40:       24420004        addiu   v0,v0,4
  400d44:       8f81801c        lw      at,-32740(gp)
  400d48:       00000000        nop
  400d4c:       24210014        addiu   at,at,20
  400d50:       ac220000        sw      v0,0(at)
  400d54:       0060c821        move    t9,v1
  400d58:       0320f809        jalr    t9
  400d5c:       00000000        nop
  400d60:       8fbc0010        lw      gp,16(sp)
  400d64:       00000000        nop
  400d68:       8f82801c        lw      v0,-32740(gp)
  400d6c:       00000000        nop
```

```
  400d70:       24420014        addiu   v0,v0,20
  400d74:       8c420000        lw      v0,0(v0)
  400d78:       00000000        nop
  400d7c:       8c430000        lw      v1,0(v0)
  400d80:       00000000        nop
  400d84:       1460ffef        bnez    v1,400d44 <__do_global_dtors_aux+0x54>
  400d88:       24420004        addiu   v0,v0,4
  400d8c:       8f828080        lw      v0,-32640(gp)
  400d90:       00000000        nop
  400d94:       10400008        beqz    v0,400db8 <__do_global_dtors_aux+0xc8>
  400d98:       00000000        nop
  400d9c:       8f84801c        lw      a0,-32740(gp)
  400da0:       00000000        nop
  400da4:       24840024        addiu   a0,a0,36
  400da8:       0040c821        move    t9,v0
  400dac:       0320f809        jalr    t9
  400db0:       00000000        nop
  400db4:       8fbc0010        lw      gp,16(sp)
  400db8:       24020001        li      v0,1
  400dbc:       8f81801c        lw      at,-32740(gp)
  400dc0:       00000000        nop
  400dc4:       24210110        addiu   at,at,272
  400dc8:       a0220000        sb      v0,0(at)
  400dcc:       8fbf001c        lw      ra,28(sp)
  400dd0:       00000000        nop
  400dd4:       03e00008        jr      ra
  400dd8:       27bd0020        addiu   sp,sp,32

00400ddc <frame_dummy>:
  400ddc:       3c1c0fc0        lui     gp,0xfc0
  400de0:       279c7254        addiu   gp,gp,29268
  400de4:       0399e021        addu    gp,gp,t9
  400de8:       27bdffe0        addiu   sp,sp,-32
  400dec:       afbc0010        sw      gp,16(sp)
  400df0:       8f828030        lw      v0,-32720(gp)
  400df4:       afbf001c        sw      ra,28(sp)
  400df8:       8f84801c        lw      a0,-32740(gp)
  400dfc:       00000000        nop
  400e00:       24840024        addiu   a0,a0,36
  400e04:       8f85801c        lw      a1,-32740(gp)
  400e08:       00000000        nop
  400e0c:       24a50114        addiu   a1,a1,276
  400e10:       10400006        beqz    v0,400e2c <frame_dummy+0x50>
  400e14:       afbc0018        sw      gp,24(sp)
  400e18:       0040c821        move    t9,v0
  400e1c:       0320f809        jalr    t9
  400e20:       00000000        nop
```

```
  400e24:       8fbc0010        lw      gp,16(sp)
  400e28:       00000000        nop
  400e2c:       8f82801c        lw      v0,-32740(gp)
  400e30:       00000000        nop
  400e34:       24420038        addiu   v0,v0,56
  400e38:       8c420000        lw      v0,0(v0)
  400e3c:       00000000        nop
  400e40:       1040000b        beqz    v0,400e70 <frame_dummy+0x94>
  400e44:       00000000        nop
  400e48:       8f828034        lw      v0,-32716(gp)
  400e4c:       8f84801c        lw      a0,-32740(gp)
  400e50:       00000000        nop
  400e54:       24840038        addiu   a0,a0,56
  400e58:       10400005        beqz    v0,400e70 <frame_dummy+0x94>
  400e5c:       00000000        nop
  400e60:       0040c821        move    t9,v0
  400e64:       0320f809        jalr    t9
  400e68:       00000000        nop
  400e6c:       8fbc0010        lw      gp,16(sp)
  400e70:       8fbf001c        lw      ra,28(sp)
  400e74:       00000000        nop
  400e78:       03e00008        jr      ra
  400e7c:       27bd0020        addiu   sp,sp,32

00400e80 <decode_byte>:
  400e80:       3c1c0fc0        lui     gp,0xfc0
  400e84:       279c71b0        addiu   gp,gp,29104
  400e88:       0399e021        addu    gp,gp,t9
  400e8c:       27bdffe8        addiu   sp,sp,-24
  400e90:       afbc0000        sw      gp,0(sp)
  400e94:       afbe0014        sw      s8,20(sp)
  400e98:       afbc0010        sw      gp,16(sp)
  400e9c:       03a0f021        move    s8,sp
  400ea0:       00801021        move    v0,a0
  400ea4:       a3c20008        sb      v0,8(s8)
  400ea8:       83c20008        lb      v0,8(s8)
  400eac:       00000000        nop
  400eb0:       28420041        slti    v0,v0,65
  400eb4:       1440000d        bnez    v0,400eec <decode_byte+0x6c>
  400eb8:       00000000        nop
  400ebc:       83c20008        lb      v0,8(s8)
  400ec0:       00000000        nop
  400ec4:       2842005b        slti    v0,v0,91
  400ec8:       10400008        beqz    v0,400eec <decode_byte+0x6c>
  400ecc:       00000000        nop
  400ed0:       93c20008        lbu     v0,8(s8)
  400ed4:       00000000        nop
```

```
400ed8:       2442ffbf        addiu   v0,v0,-65
400edc:       00021600        sll     v0,v0,0x18
400ee0:       00021603        sra     v0,v0,0x18
400ee4:       10000033        b       400fb4 <decode_byte+0x134>
400ee8:       afc2000c        sw      v0,12(s8)
400eec:       83c20008        lb      v0,8(s8)
400ef0:       00000000        nop
400ef4:       28420061        slti    v0,v0,97
400ef8:       1440000d        bnez    v0,400f30 <decode_byte+0xb0>
400efc:       00000000        nop
400f00:       83c20008        lb      v0,8(s8)
400f04:       00000000        nop
400f08:       2842007b        slti    v0,v0,123
400f0c:       10400008        beqz    v0,400f30 <decode_byte+0xb0>
400f10:       00000000        nop
400f14:       93c20008        lbu     v0,8(s8)
400f18:       00000000        nop
400f1c:       2442ffb9        addiu   v0,v0,-71
400f20:       00021600        sll     v0,v0,0x18
400f24:       00021603        sra     v0,v0,0x18
400f28:       10000022        b       400fb4 <decode_byte+0x134>
400f2c:       afc2000c        sw      v0,12(s8)
400f30:       83c20008        lb      v0,8(s8)
400f34:       00000000        nop
400f38:       28420030        slti    v0,v0,48
400f3c:       1440000d        bnez    v0,400f74 <decode_byte+0xf4>
400f40:       00000000        nop
400f44:       83c20008        lb      v0,8(s8)
400f48:       00000000        nop
400f4c:       2842003a        slti    v0,v0,58
400f50:       10400008        beqz    v0,400f74 <decode_byte+0xf4>
400f54:       00000000        nop
400f58:       93c20008        lbu     v0,8(s8)
400f5c:       00000000        nop
400f60:       24420004        addiu   v0,v0,4
400f64:       00021600        sll     v0,v0,0x18
400f68:       00021603        sra     v0,v0,0x18
400f6c:       10000011        b       400fb4 <decode_byte+0x134>
400f70:       afc2000c        sw      v0,12(s8)
400f74:       83c30008        lb      v1,8(s8)
400f78:       2402002b        li      v0,43
400f7c:       14620004        bne     v1,v0,400f90 <decode_byte+0x110>
400f80:       00000000        nop
400f84:       2402003e        li      v0,62
400f88:       1000000a        b       400fb4 <decode_byte+0x134>
400f8c:       afc2000c        sw      v0,12(s8)
400f90:       83c30008        lb      v1,8(s8)
```

```
  400f94:       2402002f        li      v0,47
  400f98:       14620004        bne     v1,v0,400fac <decode_byte+0x12c>
  400f9c:       00000000        nop
  400fa0:       2402003f        li      v0,63
  400fa4:       10000003        b       400fb4 <decode_byte+0x134>
  400fa8:       afc2000c        sw      v0,12(s8)
  400fac:       2402ffff        li      v0,-1
  400fb0:       afc2000c        sw      v0,12(s8)
  400fb4:       8fc2000c        lw      v0,12(s8)
  400fb8:       03c0e821        move    sp,s8
  400fbc:       8fbe0014        lw      s8,20(sp)
  400fc0:       03e00008        jr      ra
  400fc4:       27bd0018        addiu   sp,sp,24

00400fc8 <base64_transform_3bytes>:
  400fc8:       3c1c0fc0        lui     gp,0xfc0
  400fcc:       279c7068        addiu   gp,gp,28776
  400fd0:       0399e021        addu    gp,gp,t9
  400fd4:       27bdffc8        addiu   sp,sp,-56
  400fd8:       afbc0010        sw      gp,16(sp)
  400fdc:       afbf0030        sw      ra,48(sp)
  400fe0:       afbe002c        sw      s8,44(sp)
  400fe4:       afbc0028        sw      gp,40(sp)
  400fe8:       03a0f021        move    s8,sp
  400fec:       afc40038        sw      a0,56(s8)
  400ff0:       24040004        li      a0,4
  400ff4:       8f998098        lw      t9,-32616(gp)
  400ff8:       00000000        nop
  400ffc:       0320f809        jalr    t9
  401000:       00000000        nop
  401004:       8fdc0010        lw      gp,16(s8)
  401008:       afc20018        sw      v0,24(s8)
  40100c:       8fc20018        lw      v0,24(s8)
  401010:       00000000        nop
  401014:       14400003        bnez    v0,401024 <base64_transform_3bytes+0x5c>
  401018:       00000000        nop
  40101c:       1000008c        b       401250 <base64_transform_3bytes+0x288>
  401020:       afc00024        sw      zero,36(s8)
  401024:       afc0001c        sw      zero,28(s8)
  401028:       8fc2001c        lw      v0,28(s8)
  40102c:       00000000        nop
  401030:       28420004        slti    v0,v0,4
  401034:       14400003        bnez    v0,401044 <base64_transform_3bytes+0x7c>
  401038:       00000000        nop
  40103c:       10000081        b       401244 <base64_transform_3bytes+0x27c>
  401040:       00000000        nop
  401044:       8fc3001c        lw      v1,28(s8)
```

```
401048:       00000000        nop
40104c:       00601021        move    v0,v1
401050:       00021040        sll     v0,v0,0x1
401054:       00431021        addu    v0,v0,v1
401058:       00022840        sll     a1,v0,0x1
40105c:       8fc3001c        lw      v1,28(s8)
401060:       00000000        nop
401064:       00601021        move    v0,v1
401068:       00021040        sll     v0,v0,0x1
40106c:       00431021        addu    v0,v0,v1
401070:       00021040        sll     v0,v0,0x1
401074:       24420005        addiu   v0,v0,5
401078:       8fc40038        lw      a0,56(s8)
40107c:       00403021        move    a2,v0
401080:       8f9980a0        lw      t9,-32608(gp)
401084:       00000000        nop
401088:       0320f809        jalr    t9
40108c:       00000000        nop
401090:       8fdc0010        lw      gp,16(s8)
401094:       afc20020        sw      v0,32(s8)
401098:       8fc20020        lw      v0,32(s8)
40109c:       00000000        nop
4010a0:       14400009        bnez    v0,4010c8 <base64_transform_3bytes+0x100>
4010a4:       00000000        nop
4010a8:       8fc40018        lw      a0,24(s8)
4010ac:       8f99802c        lw      t9,-32724(gp)
4010b0:       00000000        nop
4010b4:       0320f809        jalr    t9
4010b8:       00000000        nop
4010bc:       8fdc0010        lw      gp,16(s8)
4010c0:       10000063        b       401250 <base64_transform_3bytes+0x288>
4010c4:       afc00024        sw      zero,36(s8)
4010c8:       8fc2001c        lw      v0,28(s8)
4010cc:       00000000        nop
4010d0:       1440000c        bnez    v0,401104 <base64_transform_3bytes+0x13c>
4010d4:       00000000        nop
4010d8:       8fc30018        lw      v1,24(s8)
4010dc:       8fc2001c        lw      v0,28(s8)
4010e0:       00000000        nop
4010e4:       00621821        addu    v1,v1,v0
4010e8:       8fc20020        lw      v0,32(s8)
4010ec:       00000000        nop
4010f0:       90420000        lbu     v0,0(v0)
4010f4:       00000000        nop
4010f8:       00021082        srl     v0,v0,0x2
4010fc:       10000036        b       4011d8 <base64_transform_3bytes+0x210>
401100:       a0620000        sb      v0,0(v1)
```

```
401104:       8fc3001c        lw      v1,28(s8)
401108:       24020001        li      v0,1
40110c:       14620013        bne     v1,v0,40115c <base64_transform_3bytes+0x194>
401110:       00000000        nop
401114:       8fc30018        lw      v1,24(s8)
401118:       8fc2001c        lw      v0,28(s8)
40111c:       00000000        nop
401120:       00622021        addu    a0,v1,v0
401124:       8fc20020        lw      v0,32(s8)
401128:       00000000        nop
40112c:       90420000        lbu     v0,0(v0)
401130:       00000000        nop
401134:       00021900        sll     v1,v0,0x4
401138:       8fc20020        lw      v0,32(s8)
40113c:       00000000        nop
401140:       24420001        addiu   v0,v0,1
401144:       90420000        lbu     v0,0(v0)
401148:       00000000        nop
40114c:       00021102        srl     v0,v0,0x4
401150:       00621025        or      v0,v1,v0
401154:       10000020        b       4011d8 <base64_transform_3bytes+0x210>
401158:       a0820000        sb      v0,0(a0)
40115c:       8fc3001c        lw      v1,28(s8)
401160:       24020002        li      v0,2
401164:       14620013        bne     v1,v0,4011b4 <base64_transform_3bytes+0x1ec>
401168:       00000000        nop
40116c:       8fc30018        lw      v1,24(s8)
401170:       8fc2001c        lw      v0,28(s8)
401174:       00000000        nop
401178:       00622021        addu    a0,v1,v0
40117c:       8fc20020        lw      v0,32(s8)
401180:       00000000        nop
401184:       90420000        lbu     v0,0(v0)
401188:       00000000        nop
40118c:       00021880        sll     v1,v0,0x2
401190:       8fc20020        lw      v0,32(s8)
401194:       00000000        nop
401198:       24420001        addiu   v0,v0,1
40119c:       90420000        lbu     v0,0(v0)
4011a0:       00000000        nop
4011a4:       00021182        srl     v0,v0,0x6
4011a8:       00621025        or      v0,v1,v0
4011ac:       1000000a        b       4011d8 <base64_transform_3bytes+0x210>
4011b0:       a0820000        sb      v0,0(a0)
4011b4:       8fc30018        lw      v1,24(s8)
4011b8:       8fc2001c        lw      v0,28(s8)
4011bc:       00000000        nop
```

```
  4011c0:       00621821        addu    v1,v1,v0
  4011c4:       8fc20020        lw      v0,32(s8)
  4011c8:       00000000        nop
  4011cc:       90420000        lbu     v0,0(v0)
  4011d0:       00000000        nop
  4011d4:       a0620000        sb      v0,0(v1)
  4011d8:       8fc40020        lw      a0,32(s8)
  4011dc:       8f99802c        lw      t9,-32724(gp)
  4011e0:       00000000        nop
  4011e4:       0320f809        jalr    t9
  4011e8:       00000000        nop
  4011ec:       8fdc0010        lw      gp,16(s8)
  4011f0:       8fc30018        lw      v1,24(s8)
  4011f4:       8fc2001c        lw      v0,28(s8)
  4011f8:       00000000        nop
  4011fc:       00622021        addu    a0,v1,v0
  401200:       8fc30018        lw      v1,24(s8)
  401204:       8fc2001c        lw      v0,28(s8)
  401208:       00000000        nop
  40120c:       00621021        addu    v0,v1,v0
  401210:       80420000        lb      v0,0(v0)
  401214:       8f818018        lw      at,-32744(gp)
  401218:       00000000        nop
  40121c:       24213570        addiu   at,at,13680
  401220:       00220821        addu    at,at,v0
  401224:       90220000        lbu     v0,0(at)
  401228:       00000000        nop
  40122c:       a0820000        sb      v0,0(a0)
  401230:       8fc2001c        lw      v0,28(s8)
  401234:       00000000        nop
  401238:       24420001        addiu   v0,v0,1
  40123c:       1000ff7a        b       401028 <base64_transform_3bytes+0x60>
  401240:       afc2001c        sw      v0,28(s8)
  401244:       8fc20018        lw      v0,24(s8)
  401248:       00000000        nop
  40124c:       afc20024        sw      v0,36(s8)
  401250:       8fc20024        lw      v0,36(s8)
  401254:       03c0e821        move    sp,s8
  401258:       8fbf0030        lw      ra,48(sp)
  40125c:       8fbe002c        lw      s8,44(sp)
  401260:       03e00008        jr      ra
  401264:       27bd0038        addiu   sp,sp,56

00401268 <base64_encode_bytes>:
  401268:       3c1c0fc0        lui     gp,0xfc0
  40126c:       279c6dc8        addiu   gp,gp,28104
  401270:       0399e021        addu    gp,gp,t9
```

```
401274:       27bdffb0        addiu   sp,sp,-80
401278:       afbc0010        sw      gp,16(sp)
40127c:       afbf0048        sw      ra,72(sp)
401280:       afbe0044        sw      s8,68(sp)
401284:       afbc0040        sw      gp,64(sp)
401288:       03a0f021        move    s8,sp
40128c:       afc40050        sw      a0,80(s8)
401290:       afc50054        sw      a1,84(s8)
401294:       afc60058        sw      a2,88(s8)
401298:       8fc40054        lw      a0,84(s8)
40129c:       3c025555        lui     v0,0x5555
4012a0:       34425556        ori     v0,v0,0x5556
4012a4:       00820018        mult    a0,v0
4012a8:       00001810        mfhi    v1
4012ac:       000417c3        sra     v0,a0,0x1f
4012b0:       00621823        subu    v1,v1,v0
4012b4:       00601021        move    v0,v1
4012b8:       00021040        sll     v0,v0,0x1
4012bc:       00431021        addu    v0,v0,v1
4012c0:       00821023        subu    v0,a0,v0
4012c4:       18400015        blez    v0,40131c <base64_encode_bytes+0xb4>
4012c8:       00000000        nop
4012cc:       8fc40054        lw      a0,84(s8)
4012d0:       3c025555        lui     v0,0x5555
4012d4:       34425556        ori     v0,v0,0x5556
4012d8:       00820018        mult    a0,v0
4012dc:       00001810        mfhi    v1
4012e0:       000417c3        sra     v0,a0,0x1f
4012e4:       00621823        subu    v1,v1,v0
4012e8:       00601021        move    v0,v1
4012ec:       00021040        sll     v0,v0,0x1
4012f0:       00431021        addu    v0,v0,v1
4012f4:       00821023        subu    v0,a0,v0
4012f8:       2442fffd        addiu   v0,v0,-3
4012fc:       00402021        move    a0,v0
401300:       8f998038        lw      t9,-32712(gp)
401304:       00000000        nop
401308:       0320f809        jalr    t9
40130c:       00000000        nop
401310:       8fdc0010        lw      gp,16(s8)
401314:       10000002        b       401320 <base64_encode_bytes+0xb8>
401318:       afc2003c        sw      v0,60(s8)
40131c:       afc0003c        sw      zero,60(s8)
401320:       8fc2003c        lw      v0,60(s8)
401324:       00000000        nop
401328:       afc20018        sw      v0,24(s8)
40132c:       8fc30054        lw      v1,84(s8)
```

```
401330:       8fc20018        lw      v0,24(s8)
401334:       00000000        nop
401338:       00621021        addu    v0,v1,v0
40133c:       afc2001c        sw      v0,28(s8)
401340:       8fc4001c        lw      a0,28(s8)
401344:       24050001        li      a1,1
401348:       8f998090        lw      t9,-32624(gp)
40134c:       00000000        nop
401350:       0320f809        jalr    t9
401354:       00000000        nop
401358:       8fdc0010        lw      gp,16(s8)
40135c:       afc20020        sw      v0,32(s8)
401360:       8fc20020        lw      v0,32(s8)
401364:       00000000        nop
401368:       14400004        bnez    v0,40137c <base64_encode_bytes+0x114>
40136c:       00000000        nop
401370:       2402ffff        li      v0,-1
401374:       100000a9        b       40161c <base64_encode_bytes+0x3b4>
401378:       afc20038        sw      v0,56(s8)
40137c:       8fc40020        lw      a0,32(s8)
401380:       8fc50050        lw      a1,80(s8)
401384:       8fc60054        lw      a2,84(s8)
401388:       8f99809c        lw      t9,-32612(gp)
40138c:       00000000        nop
401390:       0320f809        jalr    t9
401394:       00000000        nop
401398:       8fdc0010        lw      gp,16(s8)
40139c:       8fc20020        lw      v0,32(s8)
4013a0:       00000000        nop
4013a4:       afc20050        sw      v0,80(s8)
4013a8:       8fc2001c        lw      v0,28(s8)
4013ac:       00000000        nop
4013b0:       afc20054        sw      v0,84(s8)
4013b4:       8fc20054        lw      v0,84(s8)
4013b8:       00000000        nop
4013bc:       000210c0        sll     v0,v0,0x3
4013c0:       afc20024        sw      v0,36(s8)
4013c4:       8fc40024        lw      a0,36(s8)
4013c8:       3c022aaa        lui     v0,0x2aaa
4013cc:       3442aaab        ori     v0,v0,0xaaab
4013d0:       00820018        mult    a0,v0
4013d4:       00001810        mfhi    v1
4013d8:       000417c3        sra     v0,a0,0x1f
4013dc:       00621023        subu    v0,v1,v0
4013e0:       afc20028        sw      v0,40(s8)
4013e4:       8fc40028        lw      a0,40(s8)
4013e8:       8f998098        lw      t9,-32616(gp)
```

```
4013ec:       00000000        nop
4013f0:       0320f809        jalr    t9
4013f4:       00000000        nop
4013f8:       8fdc0010        lw      gp,16(s8)
4013fc:       afc2002c        sw      v0,44(s8)
401400:       8fc2002c        lw      v0,44(s8)
401404:       00000000        nop
401408:       14400004        bnez    v0,40141c <base64_encode_bytes+0x1b4>
40140c:       00000000        nop
401410:       2402ffff        li      v0,-1
401414:       10000081        b       40161c <base64_encode_bytes+0x3b4>
401418:       afc20038        sw      v0,56(s8)
40141c:       afc00030        sw      zero,48(s8)
401420:       8fc4001c        lw      a0,28(s8)
401424:       3c025555        lui     v0,0x5555
401428:       34425556        ori     v0,v0,0x5556
40142c:       00820018        mult    a0,v0
401430:       00001810        mfhi    v1
401434:       000417c3        sra     v0,a0,0x1f
401438:       00621823        subu    v1,v1,v0
40143c:       8fc20030        lw      v0,48(s8)
401440:       00000000        nop
401444:       0043102a        slt     v0,v0,v1
401448:       14400003        bnez    v0,401458 <base64_encode_bytes+0x1f0>
40144c:       00000000        nop
401450:       1000004d        b       401588 <base64_encode_bytes+0x320>
401454:       00000000        nop
401458:       8fc30030        lw      v1,48(s8)
40145c:       00000000        nop
401460:       00601021        move    v0,v1
401464:       00021040        sll     v0,v0,0x1
401468:       00431021        addu    v0,v0,v1
40146c:       8fc30020        lw      v1,32(s8)
401470:       00000000        nop
401474:       00431021        addu    v0,v0,v1
401478:       00402021        move    a0,v0
40147c:       8f998084        lw      t9,-32636(gp)
401480:       00000000        nop
401484:       0320f809        jalr    t9
401488:       00000000        nop
40148c:       8fdc0010        lw      gp,16(s8)
401490:       afc20034        sw      v0,52(s8)
401494:       8fc20030        lw      v0,48(s8)
401498:       00000000        nop
40149c:       00021880        sll     v1,v0,0x2
4014a0:       8fc2002c        lw      v0,44(s8)
4014a4:       00000000        nop
```

```
4014a8:       00621821        addu    v1,v1,v0
4014ac:       8fc20034        lw      v0,52(s8)
4014b0:       00000000        nop
4014b4:       90420000        lbu     v0,0(v0)
4014b8:       00000000        nop
4014bc:       a0620000        sb      v0,0(v1)
4014c0:       8fc20030        lw      v0,48(s8)
4014c4:       00000000        nop
4014c8:       00021880        sll     v1,v0,0x2
4014cc:       8fc2002c        lw      v0,44(s8)
4014d0:       00000000        nop
4014d4:       00621021        addu    v0,v1,v0
4014d8:       24430001        addiu   v1,v0,1
4014dc:       8fc20034        lw      v0,52(s8)
4014e0:       00000000        nop
4014e4:       24420001        addiu   v0,v0,1
4014e8:       90420000        lbu     v0,0(v0)
4014ec:       00000000        nop
4014f0:       a0620000        sb      v0,0(v1)
4014f4:       8fc20030        lw      v0,48(s8)
4014f8:       00000000        nop
4014fc:       00021880        sll     v1,v0,0x2
401500:       8fc2002c        lw      v0,44(s8)
401504:       00000000        nop
401508:       00621021        addu    v0,v1,v0
40150c:       24430002        addiu   v1,v0,2
401510:       8fc20034        lw      v0,52(s8)
401514:       00000000        nop
401518:       24420002        addiu   v0,v0,2
40151c:       90420000        lbu     v0,0(v0)
401520:       00000000        nop
401524:       a0620000        sb      v0,0(v1)
401528:       8fc20030        lw      v0,48(s8)
40152c:       00000000        nop
401530:       00021880        sll     v1,v0,0x2
401534:       8fc2002c        lw      v0,44(s8)
401538:       00000000        nop
40153c:       00621021        addu    v0,v1,v0
401540:       24430003        addiu   v1,v0,3
401544:       8fc20034        lw      v0,52(s8)
401548:       00000000        nop
40154c:       24420003        addiu   v0,v0,3
401550:       90420000        lbu     v0,0(v0)
401554:       00000000        nop
401558:       a0620000        sb      v0,0(v1)
40155c:       8fc40034        lw      a0,52(s8)
401560:       8f99802c        lw      t9,-32724(gp)
```

```
401564:       00000000        nop
401568:       0320f809        jalr    t9
40156c:       00000000        nop
401570:       8fdc0010        lw      gp,16(s8)
401574:       8fc20030        lw      v0,48(s8)
401578:       00000000        nop
40157c:       24420001        addiu   v0,v0,1
401580:       1000ffa7        b       401420 <base64_encode_bytes+0x1b8>
401584:       afc20030        sw      v0,48(s8)
401588:       8fc20028        lw      v0,40(s8)
40158c:       8fc30018        lw      v1,24(s8)
401590:       00000000        nop
401594:       00431023        subu    v0,v0,v1
401598:       afc20034        sw      v0,52(s8)
40159c:       8fc20034        lw      v0,52(s8)
4015a0:       8fc30028        lw      v1,40(s8)
4015a4:       00000000        nop
4015a8:       0043102a        slt     v0,v0,v1
4015ac:       14400003        bnez    v0,4015bc <base64_encode_bytes+0x354>
4015b0:       00000000        nop
4015b4:       1000000c        b       4015e8 <base64_encode_bytes+0x380>
4015b8:       00000000        nop
4015bc:       8fc3002c        lw      v1,44(s8)
4015c0:       8fc20034        lw      v0,52(s8)
4015c4:       00000000        nop
4015c8:       00621821        addu    v1,v1,v0
4015cc:       2402003d        li      v0,61
4015d0:       a0620000        sb      v0,0(v1)
4015d4:       8fc20034        lw      v0,52(s8)
4015d8:       00000000        nop
4015dc:       24420001        addiu   v0,v0,1
4015e0:       1000ffee        b       40159c <base64_encode_bytes+0x334>
4015e4:       afc20034        sw      v0,52(s8)
4015e8:       8fc40020        lw      a0,32(s8)
4015ec:       8f99802c        lw      t9,-32724(gp)
4015f0:       00000000        nop
4015f4:       0320f809        jalr    t9
4015f8:       00000000        nop
4015fc:       8fdc0010        lw      gp,16(s8)
401600:       8fc20058        lw      v0,88(s8)
401604:       8fc3002c        lw      v1,44(s8)
401608:       00000000        nop
40160c:       ac430000        sw      v1,0(v0)
401610:       8fc20028        lw      v0,40(s8)
401614:       00000000        nop
401618:       afc20038        sw      v0,56(s8)
40161c:       8fc20038        lw      v0,56(s8)
```

```
  401620:       03c0e821        move    sp,s8
  401624:       8fbf0048        lw      ra,72(sp)
  401628:       8fbe0044        lw      s8,68(sp)
  40162c:       03e00008        jr      ra
  401630:       27bd0050        addiu   sp,sp,80

00401634 <base64_decode_4bytes>:
  401634:       3c1c0fc0        lui     gp,0xfc0
  401638:       279c69fc        addiu   gp,gp,27132
  40163c:       0399e021        addu    gp,gp,t9
  401640:       27bdffb8        addiu   sp,sp,-72
  401644:       afbc0010        sw      gp,16(sp)
  401648:       afbf0040        sw      ra,64(sp)
  40164c:       afbe003c        sw      s8,60(sp)
  401650:       afbc0038        sw      gp,56(sp)
  401654:       afb10034        sw      s1,52(sp)
  401658:       afb00030        sw      s0,48(sp)
  40165c:       03a0f021        move    s8,sp
  401660:       afc40048        sw      a0,72(s8)
  401664:       afc5004c        sw      a1,76(s8)
  401668:       8fc20048        lw      v0,72(s8)
  40166c:       00000000        nop
  401670:       24420002        addiu   v0,v0,2
  401674:       80420000        lb      v0,0(v0)
  401678:       00000000        nop
  40167c:       3842003d        xori    v0,v0,0x3d
  401680:       2c430001        sltiu   v1,v0,1
  401684:       8fc20048        lw      v0,72(s8)
  401688:       00000000        nop
  40168c:       24420003        addiu   v0,v0,3
  401690:       80420000        lb      v0,0(v0)
  401694:       00000000        nop
  401698:       3842003d        xori    v0,v0,0x3d
  40169c:       2c420001        sltiu   v0,v0,1
  4016a0:       00431021        addu    v0,v0,v1
  4016a4:       afc20018        sw      v0,24(s8)
  4016a8:       24040003        li      a0,3
  4016ac:       24050001        li      a1,1
  4016b0:       8f998090        lw      t9,-32624(gp)
  4016b4:       00000000        nop
  4016b8:       0320f809        jalr    t9
  4016bc:       00000000        nop
  4016c0:       8fdc0010        lw      gp,16(s8)
  4016c4:       afc2001c        sw      v0,28(s8)
  4016c8:       8fc2001c        lw      v0,28(s8)
  4016cc:       00000000        nop
  4016d0:       14400004        bnez    v0,4016e4 <base64_decode_4bytes+0xb0>
```

```
4016d4:       00000000        nop
4016d8:       2402ffff        li      v0,-1
4016dc:       1000015c        b       401c50 <base64_decode_4bytes+0x61c>
4016e0:       afc2002c        sw      v0,44(s8)
4016e4:       24040004        li      a0,4
4016e8:       8f998098        lw      t9,-32616(gp)
4016ec:       00000000        nop
4016f0:       0320f809        jalr    t9
4016f4:       00000000        nop
4016f8:       8fdc0010        lw      gp,16(s8)
4016fc:       afc20020        sw      v0,32(s8)
401700:       8fc40020        lw      a0,32(s8)
401704:       8fc50048        lw      a1,72(s8)
401708:       24060004        li      a2,4
40170c:       8f99809c        lw      t9,-32612(gp)
401710:       00000000        nop
401714:       0320f809        jalr    t9
401718:       00000000        nop
40171c:       8fdc0010        lw      gp,16(s8)
401720:       afc00024        sw      zero,36(s8)
401724:       8fc20024        lw      v0,36(s8)
401728:       00000000        nop
40172c:       28420004        slti    v0,v0,4
401730:       14400003        bnez    v0,401740 <base64_decode_4bytes+0x10c>
401734:       00000000        nop
401738:       1000003f        b       401838 <base64_decode_4bytes+0x204>
40173c:       00000000        nop
401740:       8fc30020        lw      v1,32(s8)
401744:       8fc20024        lw      v0,36(s8)
401748:       00000000        nop
40174c:       00621021        addu    v0,v1,v0
401750:       80420000        lb      v0,0(v0)
401754:       00000000        nop
401758:       00402021        move    a0,v0
40175c:       8f99804c        lw      t9,-32692(gp)
401760:       00000000        nop
401764:       0320f809        jalr    t9
401768:       00000000        nop
40176c:       8fdc0010        lw      gp,16(s8)
401770:       00401821        move    v1,v0
401774:       2402ffff        li      v0,-1
401778:       14620018        bne     v1,v0,4017dc <base64_decode_4bytes+0x1a8>
40177c:       00000000        nop
401780:       8fc30020        lw      v1,32(s8)
401784:       8fc20024        lw      v0,36(s8)
401788:       00000000        nop
40178c:       00621021        addu    v0,v1,v0
```

```
401790:       80430000        lb      v1,0(v0)
401794:       2402003d        li      v0,61
401798:       10620010        beq     v1,v0,4017dc <base64_decode_4bytes+0x1a8>
40179c:       00000000        nop
4017a0:       8fc40020        lw      a0,32(s8)
4017a4:       8f99802c        lw      t9,-32724(gp)
4017a8:       00000000        nop
4017ac:       0320f809        jalr    t9
4017b0:       00000000        nop
4017b4:       8fdc0010        lw      gp,16(s8)
4017b8:       8fc4001c        lw      a0,28(s8)
4017bc:       8f99802c        lw      t9,-32724(gp)
4017c0:       00000000        nop
4017c4:       0320f809        jalr    t9
4017c8:       00000000        nop
4017cc:       8fdc0010        lw      gp,16(s8)
4017d0:       2402ffff        li      v0,-1
4017d4:       1000011e        b       401c50 <base64_decode_4bytes+0x61c>
4017d8:       afc2002c        sw      v0,44(s8)
4017dc:       8fc30020        lw      v1,32(s8)
4017e0:       8fc20024        lw      v0,36(s8)
4017e4:       00000000        nop
4017e8:       00621021        addu    v0,v1,v0
4017ec:       80420000        lb      v0,0(v0)
4017f0:       00000000        nop
4017f4:       00402021        move    a0,v0
4017f8:       8f99804c        lw      t9,-32692(gp)
4017fc:       00000000        nop
401800:       0320f809        jalr    t9
401804:       00000000        nop
401808:       8fdc0010        lw      gp,16(s8)
40180c:       00402021        move    a0,v0
401810:       8fc30020        lw      v1,32(s8)
401814:       8fc20024        lw      v0,36(s8)
401818:       00000000        nop
40181c:       00621021        addu    v0,v1,v0
401820:       a0440000        sb      a0,0(v0)
401824:       8fc20024        lw      v0,36(s8)
401828:       00000000        nop
40182c:       24420001        addiu   v0,v0,1
401830:       1000ffbc        b       401724 <base64_decode_4bytes+0xf0>
401834:       afc20024        sw      v0,36(s8)
401838:       afc00024        sw      zero,36(s8)
40183c:       8fc20024        lw      v0,36(s8)
401840:       00000000        nop
401844:       28420004        slti    v0,v0,4
401848:       14400003        bnez    v0,401858 <base64_decode_4bytes+0x224>
```

```
40184c:       00000000        nop
401850:       100000c0        b       401b54 <base64_decode_4bytes+0x520>
401854:       00000000        nop
401858:       8fc20024        lw      v0,36(s8)
40185c:       00000000        nop
401860:       000218c0        sll     v1,v0,0x3
401864:       8fc20024        lw      v0,36(s8)
401868:       00000000        nop
40186c:       000210c0        sll     v0,v0,0x3
401870:       24420007        addiu   v0,v0,7
401874:       8fc40020        lw      a0,32(s8)
401878:       00602821        move    a1,v1
40187c:       00403021        move    a2,v0
401880:       8f9980a0        lw      t9,-32608(gp)
401884:       00000000        nop
401888:       0320f809        jalr    t9
40188c:       00000000        nop
401890:       8fdc0010        lw      gp,16(s8)
401894:       afc20028        sw      v0,40(s8)
401898:       8fc20028        lw      v0,40(s8)
40189c:       00000000        nop
4018a0:       14400010        bnez    v0,4018e4 <base64_decode_4bytes+0x2b0>
4018a4:       00000000        nop
4018a8:       8fc4001c        lw      a0,28(s8)
4018ac:       8f99802c        lw      t9,-32724(gp)
4018b0:       00000000        nop
4018b4:       0320f809        jalr    t9
4018b8:       00000000        nop
4018bc:       8fdc0010        lw      gp,16(s8)
4018c0:       8fc40020        lw      a0,32(s8)
4018c4:       8f99802c        lw      t9,-32724(gp)
4018c8:       00000000        nop
4018cc:       0320f809        jalr    t9
4018d0:       00000000        nop
4018d4:       8fdc0010        lw      gp,16(s8)
4018d8:       2402ffff        li      v0,-1
4018dc:       100000dc        b       401c50 <base64_decode_4bytes+0x61c>
4018e0:       afc2002c        sw      v0,44(s8)
4018e4:       8fc20024        lw      v0,36(s8)
4018e8:       00000000        nop
4018ec:       14400015        bnez    v0,401944 <base64_decode_4bytes+0x310>
4018f0:       00000000        nop
4018f4:       8fd1001c        lw      s1,28(s8)
4018f8:       8fd0001c        lw      s0,28(s8)
4018fc:       8fc20028        lw      v0,40(s8)
401900:       00000000        nop
401904:       80420000        lb      v0,0(v0)
```

```
401908:       00000000        nop
40190c:       00402021        move    a0,v0
401910:       24050002        li      a1,2
401914:       24060007        li      a2,7
401918:       8f9980c0        lw      t9,-32576(gp)
40191c:       00000000        nop
401920:       0320f809        jalr    t9
401924:       00000000        nop
401928:       8fdc0010        lw      gp,16(s8)
40192c:       00021080        sll     v0,v0,0x2
401930:       92030000        lbu     v1,0(s0)
401934:       00000000        nop
401938:       00621025        or      v0,v1,v0
40193c:       1000007a        b       401b28 <base64_decode_4bytes+0x4f4>
401940:       a2220000        sb      v0,0(s1)
401944:       8fc30024        lw      v1,36(s8)
401948:       24020001        li      v0,1
40194c:       1462002c        bne     v1,v0,401a00 <base64_decode_4bytes+0x3cc>
401950:       00000000        nop
401954:       8fd1001c        lw      s1,28(s8)
401958:       8fd0001c        lw      s0,28(s8)
40195c:       8fc20028        lw      v0,40(s8)
401960:       00000000        nop
401964:       80420000        lb      v0,0(v0)
401968:       00000000        nop
40196c:       00402021        move    a0,v0
401970:       24050002        li      a1,2
401974:       24060004        li      a2,4
401978:       8f9980c0        lw      t9,-32576(gp)
40197c:       00000000        nop
401980:       0320f809        jalr    t9
401984:       00000000        nop
401988:       8fdc0010        lw      gp,16(s8)
40198c:       00021103        sra     v0,v0,0x4
401990:       92030000        lbu     v1,0(s0)
401994:       00000000        nop
401998:       00621025        or      v0,v1,v0
40199c:       a2220000        sb      v0,0(s1)
4019a0:       8fc2001c        lw      v0,28(s8)
4019a4:       00000000        nop
4019a8:       24510001        addiu   s1,v0,1
4019ac:       8fc2001c        lw      v0,28(s8)
4019b0:       00000000        nop
4019b4:       24500001        addiu   s0,v0,1
4019b8:       8fc20028        lw      v0,40(s8)
4019bc:       00000000        nop
4019c0:       80420000        lb      v0,0(v0)
```

```
4019c4:       00000000        nop
4019c8:       00402021        move    a0,v0
4019cc:       24050004        li      a1,4
4019d0:       24060007        li      a2,7
4019d4:       8f9980c0        lw      t9,-32576(gp)
4019d8:       00000000        nop
4019dc:       0320f809        jalr    t9
4019e0:       00000000        nop
4019e4:       8fdc0010        lw      gp,16(s8)
4019e8:       00021100        sll     v0,v0,0x4
4019ec:       92030000        lbu     v1,0(s0)
4019f0:       00000000        nop
4019f4:       00621025        or      v0,v1,v0
4019f8:       1000004b        b       401b28 <base64_decode_4bytes+0x4f4>
4019fc:       a2220000        sb      v0,0(s1)
401a00:       8fc30024        lw      v1,36(s8)
401a04:       24020002        li      v0,2
401a08:       14620030        bne     v1,v0,401acc <base64_decode_4bytes+0x498>
401a0c:       00000000        nop
401a10:       8fc2001c        lw      v0,28(s8)
401a14:       00000000        nop
401a18:       24510001        addiu   s1,v0,1
401a1c:       8fc2001c        lw      v0,28(s8)
401a20:       00000000        nop
401a24:       24500001        addiu   s0,v0,1
401a28:       8fc20028        lw      v0,40(s8)
401a2c:       00000000        nop
401a30:       80420000        lb      v0,0(v0)
401a34:       00000000        nop
401a38:       00402021        move    a0,v0
401a3c:       24050002        li      a1,2
401a40:       24060005        li      a2,5
401a44:       8f9980c0        lw      t9,-32576(gp)
401a48:       00000000        nop
401a4c:       0320f809        jalr    t9
401a50:       00000000        nop
401a54:       8fdc0010        lw      gp,16(s8)
401a58:       00021083        sra     v0,v0,0x2
401a5c:       92030000        lbu     v1,0(s0)
401a60:       00000000        nop
401a64:       00621025        or      v0,v1,v0
401a68:       a2220000        sb      v0,0(s1)
401a6c:       8fc2001c        lw      v0,28(s8)
401a70:       00000000        nop
401a74:       24510002        addiu   s1,v0,2
401a78:       8fc2001c        lw      v0,28(s8)
401a7c:       00000000        nop
```

```
401a80:       24500002        addiu   s0,v0,2
401a84:       8fc20028        lw      v0,40(s8)
401a88:       00000000        nop
401a8c:       80420000        lb      v0,0(v0)
401a90:       00000000        nop
401a94:       00402021        move    a0,v0
401a98:       24050005        li      a1,5
401a9c:       24060007        li      a2,7
401aa0:       8f9980c0        lw      t9,-32576(gp)
401aa4:       00000000        nop
401aa8:       0320f809        jalr    t9
401aac:       00000000        nop
401ab0:       8fdc0010        lw      gp,16(s8)
401ab4:       00021180        sll     v0,v0,0x6
401ab8:       92030000        lbu     v1,0(s0)
401abc:       00000000        nop
401ac0:       00621025        or      v0,v1,v0
401ac4:       10000018        b       401b28 <base64_decode_4bytes+0x4f4>
401ac8:       a2220000        sb      v0,0(s1)
401acc:       8fc2001c        lw      v0,28(s8)
401ad0:       00000000        nop
401ad4:       24510002        addiu   s1,v0,2
401ad8:       8fc2001c        lw      v0,28(s8)
401adc:       00000000        nop
401ae0:       24500002        addiu   s0,v0,2
401ae4:       8fc20028        lw      v0,40(s8)
401ae8:       00000000        nop
401aec:       80420000        lb      v0,0(v0)
401af0:       00000000        nop
401af4:       00402021        move    a0,v0
401af8:       24050002        li      a1,2
401afc:       24060007        li      a2,7
401b00:       8f9980c0        lw      t9,-32576(gp)
401b04:       00000000        nop
401b08:       0320f809        jalr    t9
401b0c:       00000000        nop
401b10:       8fdc0010        lw      gp,16(s8)
401b14:       00401821        move    v1,v0
401b18:       92020000        lbu     v0,0(s0)
401b1c:       00000000        nop
401b20:       00621025        or      v0,v1,v0
401b24:       a2220000        sb      v0,0(s1)
401b28:       8fc40028        lw      a0,40(s8)
401b2c:       8f99802c        lw      t9,-32724(gp)
401b30:       00000000        nop
401b34:       0320f809        jalr    t9
401b38:       00000000        nop
```

```
401b3c:       8fdc0010        lw      gp,16(s8)
401b40:       8fc20024        lw      v0,36(s8)
401b44:       00000000        nop
401b48:       24420001        addiu   v0,v0,1
401b4c:       1000ff3b        b       40183c <base64_decode_4bytes+0x208>
401b50:       afc20024        sw      v0,36(s8)
401b54:       24030003        li      v1,3
401b58:       8fc20018        lw      v0,24(s8)
401b5c:       00000000        nop
401b60:       00621023        subu    v0,v1,v0
401b64:       00402021        move    a0,v0
401b68:       8f998098        lw      t9,-32616(gp)
401b6c:       00000000        nop
401b70:       0320f809        jalr    t9
401b74:       00000000        nop
401b78:       8fdc0010        lw      gp,16(s8)
401b7c:       afc20028        sw      v0,40(s8)
401b80:       8fc20028        lw      v0,40(s8)
401b84:       00000000        nop
401b88:       14400010        bnez    v0,401bcc <base64_decode_4bytes+0x598>
401b8c:       00000000        nop
401b90:       8fc4001c        lw      a0,28(s8)
401b94:       8f99802c        lw      t9,-32724(gp)
401b98:       00000000        nop
401b9c:       0320f809        jalr    t9
401ba0:       00000000        nop
401ba4:       8fdc0010        lw      gp,16(s8)
401ba8:       8fc40020        lw      a0,32(s8)
401bac:       8f99802c        lw      t9,-32724(gp)
401bb0:       00000000        nop
401bb4:       0320f809        jalr    t9
401bb8:       00000000        nop
401bbc:       8fdc0010        lw      gp,16(s8)
401bc0:       2402ffff        li      v0,-1
401bc4:       10000022        b       401c50 <base64_decode_4bytes+0x61c>
401bc8:       afc2002c        sw      v0,44(s8)
401bcc:       24030003        li      v1,3
401bd0:       8fc20018        lw      v0,24(s8)
401bd4:       00000000        nop
401bd8:       00621023        subu    v0,v1,v0
401bdc:       8fc40028        lw      a0,40(s8)
401be0:       8fc5001c        lw      a1,28(s8)
401be4:       00403021        move    a2,v0
401be8:       8f99809c        lw      t9,-32612(gp)
401bec:       00000000        nop
401bf0:       0320f809        jalr    t9
401bf4:       00000000        nop
```

```
  401bf8:       8fdc0010        lw      gp,16(s8)
  401bfc:       8fc3004c        lw      v1,76(s8)
  401c00:       8fc20028        lw      v0,40(s8)
  401c04:       00000000        nop
  401c08:       ac620000        sw      v0,0(v1)
  401c0c:       8fc40020        lw      a0,32(s8)
  401c10:       8f99802c        lw      t9,-32724(gp)
  401c14:       00000000        nop
  401c18:       0320f809        jalr    t9
  401c1c:       00000000        nop
  401c20:       8fdc0010        lw      gp,16(s8)
  401c24:       8fc4001c        lw      a0,28(s8)
  401c28:       8f99802c        lw      t9,-32724(gp)
  401c2c:       00000000        nop
  401c30:       0320f809        jalr    t9
  401c34:       00000000        nop
  401c38:       8fdc0010        lw      gp,16(s8)
  401c3c:       24030003        li      v1,3
  401c40:       8fc20018        lw      v0,24(s8)
  401c44:       00000000        nop
  401c48:       00621023        subu    v0,v1,v0
  401c4c:       afc2002c        sw      v0,44(s8)
  401c50:       8fc2002c        lw      v0,44(s8)
  401c54:       03c0e821        move    sp,s8
  401c58:       8fbf0040        lw      ra,64(sp)
  401c5c:       8fbe003c        lw      s8,60(sp)
  401c60:       8fb10034        lw      s1,52(sp)
  401c64:       8fb00030        lw      s0,48(sp)
  401c68:       03e00008        jr      ra
  401c6c:       27bd0048        addiu   sp,sp,72

00401c70 <base64_decode_bytes>:
  401c70:       3c1c0fc0        lui     gp,0xfc0
  401c74:       279c63c0        addiu   gp,gp,25536
  401c78:       0399e021        addu    gp,gp,t9
  401c7c:       27bdffb8        addiu   sp,sp,-72
  401c80:       afbc0010        sw      gp,16(sp)
  401c84:       afbf0040        sw      ra,64(sp)
  401c88:       afbe003c        sw      s8,60(sp)
  401c8c:       afbc0038        sw      gp,56(sp)
  401c90:       03a0f021        move    s8,sp
  401c94:       afc40048        sw      a0,72(s8)
  401c98:       afc5004c        sw      a1,76(s8)
  401c9c:       afc60050        sw      a2,80(s8)
  401ca0:       afc00018        sw      zero,24(s8)
  401ca4:       8fc2004c        lw      v0,76(s8)
  401ca8:       00000000        nop
```

```
401cac:       30420003        andi    v0,v0,0x3
401cb0:       10400004        beqz    v0,401cc4 <base64_decode_bytes+0x54>
401cb4:       00000000        nop
401cb8:       2402ffff        li      v0,-1
401cbc:       10000085        b       401ed4 <base64_decode_bytes+0x264>
401cc0:       afc20030        sw      v0,48(s8)
401cc4:       afc00024        sw      zero,36(s8)
401cc8:       afc00028        sw      zero,40(s8)
401ccc:       8fc2004c        lw      v0,76(s8)
401cd0:       00000000        nop
401cd4:       afc20034        sw      v0,52(s8)
401cd8:       8fc30034        lw      v1,52(s8)
401cdc:       00000000        nop
401ce0:       04610005        bgez    v1,401cf8 <base64_decode_bytes+0x88>
401ce4:       00000000        nop
401ce8:       8fc20034        lw      v0,52(s8)
401cec:       00000000        nop
401cf0:       24420003        addiu   v0,v0,3
401cf4:       afc20034        sw      v0,52(s8)
401cf8:       8fc30034        lw      v1,52(s8)
401cfc:       00000000        nop
401d00:       00031083        sra     v0,v1,0x2
401d04:       8fc30028        lw      v1,40(s8)
401d08:       00000000        nop
401d0c:       0062102a        slt     v0,v1,v0
401d10:       14400003        bnez    v0,401d20 <base64_decode_bytes+0xb0>
401d14:       00000000        nop
401d18:       10000067        b       401eb8 <base64_decode_bytes+0x248>
401d1c:       00000000        nop
401d20:       8fc20028        lw      v0,40(s8)
401d24:       00000000        nop
401d28:       00021880        sll     v1,v0,0x2
401d2c:       8fc20048        lw      v0,72(s8)
401d30:       00000000        nop
401d34:       00621021        addu    v0,v1,v0
401d38:       27c3001c        addiu   v1,s8,28
401d3c:       00402021        move    a0,v0
401d40:       00602821        move    a1,v1
401d44:       8f9980a4        lw      t9,-32604(gp)
401d48:       00000000        nop
401d4c:       0320f809        jalr    t9
401d50:       00000000        nop
401d54:       8fdc0010        lw      gp,16(s8)
401d58:       afc20020        sw      v0,32(s8)
401d5c:       8fc20020        lw      v0,32(s8)
401d60:       00000000        nop
401d64:       0441000a        bgez    v0,401d90 <base64_decode_bytes+0x120>
```

```
401d68:       00000000        nop
401d6c:       8fc40018        lw      a0,24(s8)
401d70:       8f99802c        lw      t9,-32724(gp)
401d74:       00000000        nop
401d78:       0320f809        jalr    t9
401d7c:       00000000        nop
401d80:       8fdc0010        lw      gp,16(s8)
401d84:       2402ffff        li      v0,-1
401d88:       10000052        b       401ed4 <base64_decode_bytes+0x264>
401d8c:       afc20030        sw      v0,48(s8)
401d90:       8fc20018        lw      v0,24(s8)
401d94:       00000000        nop
401d98:       14400009        bnez    v0,401dc0 <base64_decode_bytes+0x150>
401d9c:       00000000        nop
401da0:       8fc40020        lw      a0,32(s8)
401da4:       8f998098        lw      t9,-32616(gp)
401da8:       00000000        nop
401dac:       0320f809        jalr    t9
401db0:       00000000        nop
401db4:       8fdc0010        lw      gp,16(s8)
401db8:       10000023        b       401e48 <base64_decode_bytes+0x1d8>
401dbc:       afc20018        sw      v0,24(s8)
401dc0:       8fc30024        lw      v1,36(s8)
401dc4:       8fc20020        lw      v0,32(s8)
401dc8:       00000000        nop
401dcc:       00621021        addu    v0,v1,v0
401dd0:       8fc40018        lw      a0,24(s8)
401dd4:       00402821        move    a1,v0
401dd8:       8f998078        lw      t9,-32648(gp)
401ddc:       00000000        nop
401de0:       0320f809        jalr    t9
401de4:       00000000        nop
401de8:       8fdc0010        lw      gp,16(s8)
401dec:       afc2002c        sw      v0,44(s8)
401df0:       8fc2002c        lw      v0,44(s8)
401df4:       00000000        nop
401df8:       14400010        bnez    v0,401e3c <base64_decode_bytes+0x1cc>
401dfc:       00000000        nop
401e00:       8fc40018        lw      a0,24(s8)
401e04:       8f99802c        lw      t9,-32724(gp)
401e08:       00000000        nop
401e0c:       0320f809        jalr    t9
401e10:       00000000        nop
401e14:       8fdc0010        lw      gp,16(s8)
401e18:       8fc4001c        lw      a0,28(s8)
401e1c:       8f99802c        lw      t9,-32724(gp)
401e20:       00000000        nop
```

```
401e24:       0320f809        jalr    t9
401e28:       00000000        nop
401e2c:       8fdc0010        lw      gp,16(s8)
401e30:       2403ffff        li      v1,-1
401e34:       10000027        b       401ed4 <base64_decode_bytes+0x264>
401e38:       afc30030        sw      v1,48(s8)
401e3c:       8fc2002c        lw      v0,44(s8)
401e40:       00000000        nop
401e44:       afc20018        sw      v0,24(s8)
401e48:       8fc30018        lw      v1,24(s8)
401e4c:       8fc20024        lw      v0,36(s8)
401e50:       00000000        nop
401e54:       00621021        addu    v0,v1,v0
401e58:       00402021        move    a0,v0
401e5c:       8fc5001c        lw      a1,28(s8)
401e60:       8fc60020        lw      a2,32(s8)
401e64:       8f99809c        lw      t9,-32612(gp)
401e68:       00000000        nop
401e6c:       0320f809        jalr    t9
401e70:       00000000        nop
401e74:       8fdc0010        lw      gp,16(s8)
401e78:       8fc30024        lw      v1,36(s8)
401e7c:       8fc20020        lw      v0,32(s8)
401e80:       00000000        nop
401e84:       00621021        addu    v0,v1,v0
401e88:       afc20024        sw      v0,36(s8)
401e8c:       8fc4001c        lw      a0,28(s8)
401e90:       8f99802c        lw      t9,-32724(gp)
401e94:       00000000        nop
401e98:       0320f809        jalr    t9
401e9c:       00000000        nop
401ea0:       8fdc0010        lw      gp,16(s8)
401ea4:       8fc20028        lw      v0,40(s8)
401ea8:       00000000        nop
401eac:       24420001        addiu   v0,v0,1
401eb0:       1000ff86        b       401ccc <base64_decode_bytes+0x5c>
401eb4:       afc20028        sw      v0,40(s8)
401eb8:       8fc20050        lw      v0,80(s8)
401ebc:       8fc30018        lw      v1,24(s8)
401ec0:       00000000        nop
401ec4:       ac430000        sw      v1,0(v0)
401ec8:       8fc20024        lw      v0,36(s8)
401ecc:       00000000        nop
401ed0:       afc20030        sw      v0,48(s8)
401ed4:       8fc20030        lw      v0,48(s8)
401ed8:       03c0e821        move    sp,s8
401edc:       8fbf0040        lw      ra,64(sp)
```

```
  401ee0:       8fbe003c        lw      s8,60(sp)
  401ee4:       03e00008        jr      ra
  401ee8:       27bd0048        addiu   sp,sp,72
  401eec:       00000000        nop

00401ef0 <byte_nth_bit>:
  401ef0:       3c1c0fc0        lui     gp,0xfc0
  401ef4:       279c6140        addiu   gp,gp,24896
  401ef8:       0399e021        addu    gp,gp,t9
  401efc:       27bdffd0        addiu   sp,sp,-48
  401f00:       afbc0010        sw      gp,16(sp)
  401f04:       afbf0028        sw      ra,40(sp)
  401f08:       afbe0024        sw      s8,36(sp)
  401f0c:       afbc0020        sw      gp,32(sp)
  401f10:       03a0f021        move    s8,sp
  401f14:       00801021        move    v0,a0
  401f18:       afc50034        sw      a1,52(s8)
  401f1c:       a3c20018        sb      v0,24(s8)
  401f20:       8fc20034        lw      v0,52(s8)
  401f24:       00000000        nop
  401f28:       2442fff9        addiu   v0,v0,-7
  401f2c:       00402021        move    a0,v0
  401f30:       8f998038        lw      t9,-32712(gp)
  401f34:       00000000        nop
  401f38:       0320f809        jalr    t9
  401f3c:       00000000        nop
  401f40:       8fdc0010        lw      gp,16(s8)
  401f44:       afc20034        sw      v0,52(s8)
  401f48:       24020001        li      v0,1
  401f4c:       a3c20019        sb      v0,25(s8)
  401f50:       83c30018        lb      v1,24(s8)
  401f54:       8fc20034        lw      v0,52(s8)
  401f58:       00000000        nop
  401f5c:       00431007        srav    v0,v1,v0
  401f60:       a3c20018        sb      v0,24(s8)
  401f64:       93c30019        lbu     v1,25(s8)
  401f68:       93c20018        lbu     v0,24(s8)
  401f6c:       00000000        nop
  401f70:       00621024        and     v0,v1,v0
  401f74:       00021600        sll     v0,v0,0x18
  401f78:       00021603        sra     v0,v0,0x18
  401f7c:       0002102b        sltu    v0,zero,v0
  401f80:       03c0e821        move    sp,s8
  401f84:       8fbf0028        lw      ra,40(sp)
  401f88:       8fbe0024        lw      s8,36(sp)
  401f8c:       03e00008        jr      ra
  401f90:       27bd0030        addiu   sp,sp,48
```

```
00401f94 <byte_mask_nth_bit>:
  401f94:       3c1c0fc0        lui     gp,0xfc0
  401f98:       279c609c        addiu   gp,gp,24732
  401f9c:       0399e021        addu    gp,gp,t9
  401fa0:       27bdffd0        addiu   sp,sp,-48
  401fa4:       afbc0010        sw      gp,16(sp)
  401fa8:       afbf0028        sw      ra,40(sp)
  401fac:       afbe0024        sw      s8,36(sp)
  401fb0:       afbc0020        sw      gp,32(sp)
  401fb4:       03a0f021        move    s8,sp
  401fb8:       00801021        move    v0,a0
  401fbc:       afc50034        sw      a1,52(s8)
  401fc0:       a3c20018        sb      v0,24(s8)
  401fc4:       83c20018        lb      v0,24(s8)
  401fc8:       00000000        nop
  401fcc:       00402021        move    a0,v0
  401fd0:       8fc50034        lw      a1,52(s8)
  401fd4:       8f9980ac        lw      t9,-32596(gp)
  401fd8:       00000000        nop
  401fdc:       0320f809        jalr    t9
  401fe0:       00000000        nop
  401fe4:       8fdc0010        lw      gp,16(s8)
  401fe8:       a3c20019        sb      v0,25(s8)
  401fec:       8fc20034        lw      v0,52(s8)
  401ff0:       00000000        nop
  401ff4:       2442fff9        addiu   v0,v0,-7
  401ff8:       00402021        move    a0,v0
  401ffc:       8f998038        lw      t9,-32712(gp)
  402000:       00000000        nop
  402004:       0320f809        jalr    t9
  402008:       00000000        nop
  40200c:       8fdc0010        lw      gp,16(s8)
  402010:       afc20034        sw      v0,52(s8)
  402014:       83c30019        lb      v1,25(s8)
  402018:       8fc20034        lw      v0,52(s8)
  40201c:       00000000        nop
  402020:       00431004        sllv    v0,v1,v0
  402024:       00021600        sll     v0,v0,0x18
  402028:       00021603        sra     v0,v0,0x18
  40202c:       03c0e821        move    sp,s8
  402030:       8fbf0028        lw      ra,40(sp)
  402034:       8fbe0024        lw      s8,36(sp)
  402038:       03e00008        jr      ra
  40203c:       27bd0030        addiu   sp,sp,48

00402040 <byte_extract_n_bits>:
```

```
402040:       3c1c0fc0        lui     gp,0xfc0
402044:       279c5ff0        addiu   gp,gp,24560
402048:       0399e021        addu    gp,gp,t9
40204c:       27bdffd0        addiu   sp,sp,-48
402050:       afbc0010        sw      gp,16(sp)
402054:       afbf0028        sw      ra,40(sp)
402058:       afbe0024        sw      s8,36(sp)
40205c:       afbc0020        sw      gp,32(sp)
402060:       03a0f021        move    s8,sp
402064:       00801021        move    v0,a0
402068:       afc50034        sw      a1,52(s8)
40206c:       afc60038        sw      a2,56(s8)
402070:       a3c20018        sb      v0,24(s8)
402074:       a3c00019        sb      zero,25(s8)
402078:       8fc20034        lw      v0,52(s8)
40207c:       00000000        nop
402080:       afc2001c        sw      v0,28(s8)
402084:       8fc2001c        lw      v0,28(s8)
402088:       8fc30038        lw      v1,56(s8)
40208c:       00000000        nop
402090:       0062102a        slt     v0,v1,v0
402094:       10400003        beqz    v0,4020a4 <byte_extract_n_bits+0x64>
402098:       00000000        nop
40209c:       10000014        b       4020f0 <byte_extract_n_bits+0xb0>
4020a0:       00000000        nop
4020a4:       83c20018        lb      v0,24(s8)
4020a8:       00000000        nop
4020ac:       00402021        move    a0,v0
4020b0:       8fc5001c        lw      a1,28(s8)
4020b4:       8f998050        lw      t9,-32688(gp)
4020b8:       00000000        nop
4020bc:       0320f809        jalr    t9
4020c0:       00000000        nop
4020c4:       8fdc0010        lw      gp,16(s8)
4020c8:       00401821        move    v1,v0
4020cc:       93c20019        lbu     v0,25(s8)
4020d0:       00000000        nop
4020d4:       00621025        or      v0,v1,v0
4020d8:       a3c20019        sb      v0,25(s8)
4020dc:       8fc2001c        lw      v0,28(s8)
4020e0:       00000000        nop
4020e4:       24420001        addiu   v0,v0,1
4020e8:       1000ffe6        b       402084 <byte_extract_n_bits+0x44>
4020ec:       afc2001c        sw      v0,28(s8)
4020f0:       83c20019        lb      v0,25(s8)
4020f4:       03c0e821        move    sp,s8
4020f8:       8fbf0028        lw      ra,40(sp)
```

```
  4020fc:       8fbe0024        lw      s8,36(sp)
  402100:       03e00008        jr      ra
  402104:       27bd0030        addiu   sp,sp,48

00402108 <bytes_nth_bit>:
  402108:       3c1c0fc0        lui     gp,0xfc0
  40210c:       279c5f28        addiu   gp,gp,24360
  402110:       0399e021        addu    gp,gp,t9
  402114:       27bdffc8        addiu   sp,sp,-56
  402118:       afbc0010        sw      gp,16(sp)
  40211c:       afbf0030        sw      ra,48(sp)
  402120:       afbe002c        sw      s8,44(sp)
  402124:       afbc0028        sw      gp,40(sp)
  402128:       03a0f021        move    s8,sp
  40212c:       afc40038        sw      a0,56(s8)
  402130:       afc5003c        sw      a1,60(s8)
  402134:       8fc2003c        lw      v0,60(s8)
  402138:       00000000        nop
  40213c:       afc20020        sw      v0,32(s8)
  402140:       8fc20020        lw      v0,32(s8)
  402144:       00000000        nop
  402148:       04410005        bgez    v0,402160 <bytes_nth_bit+0x58>
  40214c:       00000000        nop
  402150:       8fc30020        lw      v1,32(s8)
  402154:       00000000        nop
  402158:       24630007        addiu   v1,v1,7
  40215c:       afc30020        sw      v1,32(s8)
  402160:       8fc30020        lw      v1,32(s8)
  402164:       00000000        nop
  402168:       000310c3        sra     v0,v1,0x3
  40216c:       afc20018        sw      v0,24(s8)
  402170:       8fc20018        lw      v0,24(s8)
  402174:       00000000        nop
  402178:       000218c0        sll     v1,v0,0x3
  40217c:       8fc2003c        lw      v0,60(s8)
  402180:       00000000        nop
  402184:       00431023        subu    v0,v0,v1
  402188:       afc2001c        sw      v0,28(s8)
  40218c:       8fc30038        lw      v1,56(s8)
  402190:       8fc20018        lw      v0,24(s8)
  402194:       00000000        nop
  402198:       00621021        addu    v0,v1,v0
  40219c:       80420000        lb      v0,0(v0)
  4021a0:       00000000        nop
  4021a4:       00402021        move    a0,v0
  4021a8:       8fc5001c        lw      a1,28(s8)
  4021ac:       8f9980ac        lw      t9,-32596(gp)
```

```
  4021b0:       00000000        nop
  4021b4:       0320f809        jalr    t9
  4021b8:       00000000        nop
  4021bc:       8fdc0010        lw      gp,16(s8)
  4021c0:       03c0e821        move    sp,s8
  4021c4:       8fbf0030        lw      ra,48(sp)
  4021c8:       8fbe002c        lw      s8,44(sp)
  4021cc:       03e00008        jr      ra
  4021d0:       27bd0038        addiu   sp,sp,56

004021d4 <bytes_mask_nth_bit>:
  4021d4:       3c1c0fc0        lui     gp,0xfc0
  4021d8:       279c5e5c        addiu   gp,gp,24156
  4021dc:       0399e021        addu    gp,gp,t9
  4021e0:       27bdffc8        addiu   sp,sp,-56
  4021e4:       afbc0010        sw      gp,16(sp)
  4021e8:       afbf0030        sw      ra,48(sp)
  4021ec:       afbe002c        sw      s8,44(sp)
  4021f0:       afbc0028        sw      gp,40(sp)
  4021f4:       03a0f021        move    s8,sp
  4021f8:       afc40038        sw      a0,56(s8)
  4021fc:       afc5003c        sw      a1,60(s8)
  402200:       8fc2003c        lw      v0,60(s8)
  402204:       00000000        nop
  402208:       afc20020        sw      v0,32(s8)
  40220c:       8fc20020        lw      v0,32(s8)
  402210:       00000000        nop
  402214:       04410005        bgez    v0,40222c <bytes_mask_nth_bit+0x58>
  402218:       00000000        nop
  40221c:       8fc30020        lw      v1,32(s8)
  402220:       00000000        nop
  402224:       24630007        addiu   v1,v1,7
  402228:       afc30020        sw      v1,32(s8)
  40222c:       8fc30020        lw      v1,32(s8)
  402230:       00000000        nop
  402234:       000310c3        sra     v0,v1,0x3
  402238:       afc20018        sw      v0,24(s8)
  40223c:       8fc3003c        lw      v1,60(s8)
  402240:       8fc20018        lw      v0,24(s8)
  402244:       00000000        nop
  402248:       00621023        subu    v0,v1,v0
  40224c:       afc2001c        sw      v0,28(s8)
  402250:       8fc30038        lw      v1,56(s8)
  402254:       8fc20018        lw      v0,24(s8)
  402258:       00000000        nop
  40225c:       00621021        addu    v0,v1,v0
  402260:       80420000        lb      v0,0(v0)
```

```
  402264:       00000000        nop
  402268:       00402021        move    a0,v0
  40226c:       8fc5001c        lw      a1,28(s8)
  402270:       8f998050        lw      t9,-32688(gp)
  402274:       00000000        nop
  402278:       0320f809        jalr    t9
  40227c:       00000000        nop
  402280:       8fdc0010        lw      gp,16(s8)
  402284:       03c0e821        move    sp,s8
  402288:       8fbf0030        lw      ra,48(sp)
  40228c:       8fbe002c        lw      s8,44(sp)
  402290:       03e00008        jr      ra
  402294:       27bd0038        addiu   sp,sp,56

00402298 <bytes_extract_n_bits>:
  402298:       3c1c0fc0        lui     gp,0xfc0
  40229c:       279c5d98        addiu   gp,gp,23960
  4022a0:       0399e021        addu    gp,gp,t9
  4022a4:       27bdffa0        addiu   sp,sp,-96
  4022a8:       afbc0010        sw      gp,16(sp)
  4022ac:       afbf0058        sw      ra,88(sp)
  4022b0:       afbe0054        sw      s8,84(sp)
  4022b4:       afbc0050        sw      gp,80(sp)
  4022b8:       03a0f021        move    s8,sp
  4022bc:       afc40060        sw      a0,96(s8)
  4022c0:       afc50064        sw      a1,100(s8)
  4022c4:       afc60068        sw      a2,104(s8)
  4022c8:       8fc20064        lw      v0,100(s8)
  4022cc:       00000000        nop
  4022d0:       afc20034        sw      v0,52(s8)
  4022d4:       8fc20034        lw      v0,52(s8)
  4022d8:       00000000        nop
  4022dc:       04410005        bgez    v0,4022f4 <bytes_extract_n_bits+0x5c>
  4022e0:       00000000        nop
  4022e4:       8fc30034        lw      v1,52(s8)
  4022e8:       00000000        nop
  4022ec:       24630007        addiu   v1,v1,7
  4022f0:       afc30034        sw      v1,52(s8)
  4022f4:       8fc30034        lw      v1,52(s8)
  4022f8:       00000000        nop
  4022fc:       000310c3        sra     v0,v1,0x3
  402300:       afc20018        sw      v0,24(s8)
  402304:       8fc20068        lw      v0,104(s8)
  402308:       00000000        nop
  40230c:       afc20038        sw      v0,56(s8)
  402310:       8fc20038        lw      v0,56(s8)
  402314:       00000000        nop
```

```
402318:       04410005        bgez    v0,402330 <bytes_extract_n_bits+0x98>
40231c:       00000000        nop
402320:       8fc30038        lw      v1,56(s8)
402324:       00000000        nop
402328:       24630007        addiu   v1,v1,7
40232c:       afc30038        sw      v1,56(s8)
402330:       8fc30038        lw      v1,56(s8)
402334:       00000000        nop
402338:       000310c3        sra     v0,v1,0x3
40233c:       afc2001c        sw      v0,28(s8)
402340:       8fc3001c        lw      v1,28(s8)
402344:       8fc20018        lw      v0,24(s8)
402348:       00000000        nop
40234c:       00621023        subu    v0,v1,v0
402350:       24420001        addiu   v0,v0,1
402354:       00402021        move    a0,v0
402358:       8f998098        lw      t9,-32616(gp)
40235c:       00000000        nop
402360:       0320f809        jalr    t9
402364:       00000000        nop
402368:       8fdc0010        lw      gp,16(s8)
40236c:       afc20020        sw      v0,32(s8)
402370:       8fc20020        lw      v0,32(s8)
402374:       00000000        nop
402378:       14400003        bnez    v0,402388 <bytes_extract_n_bits+0xf0>
40237c:       00000000        nop
402380:       10000064        b       402514 <bytes_extract_n_bits+0x27c>
402384:       afc00030        sw      zero,48(s8)
402388:       8fc20018        lw      v0,24(s8)
40238c:       00000000        nop
402390:       afc20024        sw      v0,36(s8)
402394:       8fc20024        lw      v0,36(s8)
402398:       8fc3001c        lw      v1,28(s8)
40239c:       00000000        nop
4023a0:       0062102a        slt     v0,v1,v0
4023a4:       10400003        beqz    v0,4023b4 <bytes_extract_n_bits+0x11c>
4023a8:       00000000        nop
4023ac:       10000056        b       402508 <bytes_extract_n_bits+0x270>
4023b0:       00000000        nop
4023b4:       afc00028        sw      zero,40(s8)
4023b8:       8fc30024        lw      v1,36(s8)
4023bc:       8fc20018        lw      v0,24(s8)
4023c0:       00000000        nop
4023c4:       14620017        bne     v1,v0,402424 <bytes_extract_n_bits+0x18c>
4023c8:       00000000        nop
4023cc:       8fc20064        lw      v0,100(s8)
4023d0:       00000000        nop
```

```
4023d4:       afc2003c        sw      v0,60(s8)
4023d8:       8fc3003c        lw      v1,60(s8)
4023dc:       00000000        nop
4023e0:       afc30040        sw      v1,64(s8)
4023e4:       8fc20040        lw      v0,64(s8)
4023e8:       00000000        nop
4023ec:       04410005        bgez    v0,402404 <bytes_extract_n_bits+0x16c>
4023f0:       00000000        nop
4023f4:       8fc30040        lw      v1,64(s8)
4023f8:       00000000        nop
4023fc:       24630007        addiu   v1,v1,7
402400:       afc30040        sw      v1,64(s8)
402404:       8fc30040        lw      v1,64(s8)
402408:       00000000        nop
40240c:       000310c3        sra     v0,v1,0x3
402410:       000210c0        sll     v0,v0,0x3
402414:       8fc3003c        lw      v1,60(s8)
402418:       00000000        nop
40241c:       00621023        subu    v0,v1,v0
402420:       afc20028        sw      v0,40(s8)
402424:       24020007        li      v0,7
402428:       afc2002c        sw      v0,44(s8)
40242c:       8fc30024        lw      v1,36(s8)
402430:       8fc2001c        lw      v0,28(s8)
402434:       00000000        nop
402438:       14620017        bne     v1,v0,402498 <bytes_extract_n_bits+0x200>
40243c:       00000000        nop
402440:       8fc20068        lw      v0,104(s8)
402444:       00000000        nop
402448:       afc20044        sw      v0,68(s8)
40244c:       8fc30044        lw      v1,68(s8)
402450:       00000000        nop
402454:       afc30048        sw      v1,72(s8)
402458:       8fc20048        lw      v0,72(s8)
40245c:       00000000        nop
402460:       04410005        bgez    v0,402478 <bytes_extract_n_bits+0x1e0>
402464:       00000000        nop
402468:       8fc30048        lw      v1,72(s8)
40246c:       00000000        nop
402470:       24630007        addiu   v1,v1,7
402474:       afc30048        sw      v1,72(s8)
402478:       8fc30048        lw      v1,72(s8)
40247c:       00000000        nop
402480:       000310c3        sra     v0,v1,0x3
402484:       000210c0        sll     v0,v0,0x3
402488:       8fc30044        lw      v1,68(s8)
40248c:       00000000        nop
```

```
  402490:       00621023        subu    v0,v1,v0
  402494:       afc2002c        sw      v0,44(s8)
  402498:       8fc30060        lw      v1,96(s8)
  40249c:       8fc20024        lw      v0,36(s8)
  4024a0:       00000000        nop
  4024a4:       00621021        addu    v0,v1,v0
  4024a8:       80420000        lb      v0,0(v0)
  4024ac:       00000000        nop
  4024b0:       00402021        move    a0,v0
  4024b4:       8fc50028        lw      a1,40(s8)
  4024b8:       8fc6002c        lw      a2,44(s8)
  4024bc:       8f9980c0        lw      t9,-32576(gp)
  4024c0:       00000000        nop
  4024c4:       0320f809        jalr    t9
  4024c8:       00000000        nop
  4024cc:       8fdc0010        lw      gp,16(s8)
  4024d0:       00402021        move    a0,v0
  4024d4:       8fc30024        lw      v1,36(s8)
  4024d8:       8fc20018        lw      v0,24(s8)
  4024dc:       00000000        nop
  4024e0:       00621823        subu    v1,v1,v0
  4024e4:       8fc20020        lw      v0,32(s8)
  4024e8:       00000000        nop
  4024ec:       00621021        addu    v0,v1,v0
  4024f0:       a0440000        sb      a0,0(v0)
  4024f4:       8fc20024        lw      v0,36(s8)
  4024f8:       00000000        nop
  4024fc:       24420001        addiu   v0,v0,1
  402500:       1000ffa4        b       402394 <bytes_extract_n_bits+0xfc>
  402504:       afc20024        sw      v0,36(s8)
  402508:       8fc20020        lw      v0,32(s8)
  40250c:       00000000        nop
  402510:       afc20030        sw      v0,48(s8)
  402514:       8fc20030        lw      v0,48(s8)
  402518:       03c0e821        move    sp,s8
  40251c:       8fbf0058        lw      ra,88(sp)
  402520:       8fbe0054        lw      s8,84(sp)
  402524:       03e00008        jr      ra
  402528:       27bd0060        addiu   sp,sp,96
  40252c:       00000000        nop

00402530 <imprimir_ayuda>:
  402530:       3c1c0fc0        lui     gp,0xfc0
  402534:       279c5b00        addiu   gp,gp,23296
  402538:       0399e021        addu    gp,gp,t9
  40253c:       27bdffd8        addiu   sp,sp,-40
  402540:       afbc0010        sw      gp,16(sp)
```

```
  402544:       afbf0020        sw      ra,32(sp)
  402548:       afbe001c        sw      s8,28(sp)
  40254c:       afbc0018        sw      gp,24(sp)
  402550:       03a0f021        move    s8,sp
  402554:       8f848018        lw      a0,-32744(gp)
  402558:       00000000        nop
  40255c:       248435b0        addiu   a0,a0,13744
  402560:       8f9980b8        lw      t9,-32584(gp)
  402564:       00000000        nop
  402568:       0320f809        jalr    t9
  40256c:       00000000        nop
  402570:       8fdc0010        lw      gp,16(s8)
  402574:       03c0e821        move    sp,s8
  402578:       8fbf0020        lw      ra,32(sp)
  40257c:       8fbe001c        lw      s8,28(sp)
  402580:       03e00008        jr      ra
  402584:       27bd0028        addiu   sp,sp,40

00402588 <encode_file>:
  402588:       3c1c0fc0        lui     gp,0xfc0
  40258c:       279c5aa8        addiu   gp,gp,23208
  402590:       0399e021        addu    gp,gp,t9
  402594:       27bdffb8        addiu   sp,sp,-72
  402598:       afbc0010        sw      gp,16(sp)
  40259c:       afbf0040        sw      ra,64(sp)
  4025a0:       afbe003c        sw      s8,60(sp)
  4025a4:       afbc0038        sw      gp,56(sp)
  4025a8:       03a0f021        move    s8,sp
  4025ac:       afc40048        sw      a0,72(s8)
  4025b0:       afc5004c        sw      a1,76(s8)
  4025b4:       a3c00018        sb      zero,24(s8)
  4025b8:       93c20018        lbu     v0,24(s8)
  4025bc:       00000000        nop
  4025c0:       10400003        beqz    v0,4025d0 <encode_file+0x48>
  4025c4:       00000000        nop
  4025c8:       10000066        b       402764 <encode_file+0x1dc>
  4025cc:       00000000        nop
  4025d0:       24040078        li      a0,120
  4025d4:       8f998098        lw      t9,-32616(gp)
  4025d8:       00000000        nop
  4025dc:       0320f809        jalr    t9
  4025e0:       00000000        nop
  4025e4:       8fdc0010        lw      gp,16(s8)
  4025e8:       afc20028        sw      v0,40(s8)
  4025ec:       8fc20028        lw      v0,40(s8)
  4025f0:       00000000        nop
  4025f4:       14400004        bnez    v0,402608 <encode_file+0x80>
```

```
4025f8:       00000000        nop
4025fc:       2402ffff        li      v0,-1
402600:       10000059        b       402768 <encode_file+0x1e0>
402604:       afc20030        sw      v0,48(s8)
402608:       afc0002c        sw      zero,44(s8)
40260c:       8fc2002c        lw      v0,44(s8)
402610:       00000000        nop
402614:       28420078        slti    v0,v0,120
402618:       1040001a        beqz    v0,402684 <encode_file+0xfc>
40261c:       00000000        nop
402620:       8fc40048        lw      a0,72(s8)
402624:       8f998058        lw      t9,-32680(gp)
402628:       00000000        nop
40262c:       0320f809        jalr    t9
402630:       00000000        nop
402634:       8fdc0010        lw      gp,16(s8)
402638:       afc20020        sw      v0,32(s8)
40263c:       8fc30020        lw      v1,32(s8)
402640:       2402ffff        li      v0,-1
402644:       14620003        bne     v1,v0,402654 <encode_file+0xcc>
402648:       00000000        nop
40264c:       1000000d        b       402684 <encode_file+0xfc>
402650:       00000000        nop
402654:       8fc30028        lw      v1,40(s8)
402658:       8fc2002c        lw      v0,44(s8)
40265c:       00000000        nop
402660:       00621821        addu    v1,v1,v0
402664:       93c20020        lbu     v0,32(s8)
402668:       00000000        nop
40266c:       a0620000        sb      v0,0(v1)
402670:       8fc2002c        lw      v0,44(s8)
402674:       00000000        nop
402678:       24420001        addiu   v0,v0,1
40267c:       1000ffe3        b       40260c <encode_file+0x84>
402680:       afc2002c        sw      v0,44(s8)
402684:       8fc30020        lw      v1,32(s8)
402688:       2402ffff        li      v0,-1
40268c:       14620003        bne     v1,v0,40269c <encode_file+0x114>
402690:       00000000        nop
402694:       24020001        li      v0,1
402698:       a3c20018        sb      v0,24(s8)
40269c:       8fc2002c        lw      v0,44(s8)
4026a0:       00000000        nop
4026a4:       10400027        beqz    v0,402744 <encode_file+0x1bc>
4026a8:       00000000        nop
4026ac:       27c20024        addiu   v0,s8,36
4026b0:       8fc40028        lw      a0,40(s8)
```

```
4026b4:       8fc5002c        lw      a1,44(s8)
4026b8:       00403021        move    a2,v0
4026bc:       8f9980b4        lw      t9,-32588(gp)
4026c0:       00000000        nop
4026c4:       0320f809        jalr    t9
4026c8:       00000000        nop
4026cc:       8fdc0010        lw      gp,16(s8)
4026d0:       afc2001c        sw      v0,28(s8)
4026d4:       8fc2001c        lw      v0,28(s8)
4026d8:       00000000        nop
4026dc:       0441000a        bgez    v0,402708 <encode_file+0x180>
4026e0:       00000000        nop
4026e4:       8fc40028        lw      a0,40(s8)
4026e8:       8f99802c        lw      t9,-32724(gp)
4026ec:       00000000        nop
4026f0:       0320f809        jalr    t9
4026f4:       00000000        nop
4026f8:       8fdc0010        lw      gp,16(s8)
4026fc:       2402ffff        li      v0,-1
402700:       10000019        b       402768 <encode_file+0x1e0>
402704:       afc20030        sw      v0,48(s8)
402708:       8fc40024        lw      a0,36(s8)
40270c:       24050001        li      a1,1
402710:       8fc6001c        lw      a2,28(s8)
402714:       8fc7004c        lw      a3,76(s8)
402718:       8f998048        lw      t9,-32696(gp)
40271c:       00000000        nop
402720:       0320f809        jalr    t9
402724:       00000000        nop
402728:       8fdc0010        lw      gp,16(s8)
40272c:       8fc40024        lw      a0,36(s8)
402730:       8f99802c        lw      t9,-32724(gp)
402734:       00000000        nop
402738:       0320f809        jalr    t9
40273c:       00000000        nop
402740:       8fdc0010        lw      gp,16(s8)
402744:       8fc40028        lw      a0,40(s8)
402748:       8f99802c        lw      t9,-32724(gp)
40274c:       00000000        nop
402750:       0320f809        jalr    t9
402754:       00000000        nop
402758:       8fdc0010        lw      gp,16(s8)
40275c:       1000ff96        b       4025b8 <encode_file+0x30>
402760:       00000000        nop
402764:       afc00030        sw      zero,48(s8)
402768:       8fc20030        lw      v0,48(s8)
40276c:       03c0e821        move    sp,s8
```

```
  402770:       8fbf0040        lw      ra,64(sp)
  402774:       8fbe003c        lw      s8,60(sp)
  402778:       03e00008        jr      ra
  40277c:       27bd0048        addiu   sp,sp,72

00402780 <decode_file>:
  402780:       3c1c0fc0        lui     gp,0xfc0
  402784:       279c58b0        addiu   gp,gp,22704
  402788:       0399e021        addu    gp,gp,t9
  40278c:       27bdffb8        addiu   sp,sp,-72
  402790:       afbc0010        sw      gp,16(sp)
  402794:       afbf0040        sw      ra,64(sp)
  402798:       afbe003c        sw      s8,60(sp)
  40279c:       afbc0038        sw      gp,56(sp)
  4027a0:       03a0f021        move    s8,sp
  4027a4:       afc40048        sw      a0,72(s8)
  4027a8:       afc5004c        sw      a1,76(s8)
  4027ac:       a3c00018        sb      zero,24(s8)
  4027b0:       93c20018        lbu     v0,24(s8)
  4027b4:       00000000        nop
  4027b8:       10400003        beqz    v0,4027c8 <decode_file+0x48>
  4027bc:       00000000        nop
  4027c0:       10000060        b       402944 <decode_file+0x1c4>
  4027c4:       00000000        nop
  4027c8:       24040078        li      a0,120
  4027cc:       8f998098        lw      t9,-32616(gp)
  4027d0:       00000000        nop
  4027d4:       0320f809        jalr    t9
  4027d8:       00000000        nop
  4027dc:       8fdc0010        lw      gp,16(s8)
  4027e0:       afc20020        sw      v0,32(s8)
  4027e4:       8fc20020        lw      v0,32(s8)
  4027e8:       00000000        nop
  4027ec:       14400004        bnez    v0,402800 <decode_file+0x80>
  4027f0:       00000000        nop
  4027f4:       2402ffff        li      v0,-1
  4027f8:       10000053        b       402948 <decode_file+0x1c8>
  4027fc:       afc20034        sw      v0,52(s8)
  402800:       afc00030        sw      zero,48(s8)
  402804:       8fc20030        lw      v0,48(s8)
  402808:       00000000        nop
  40280c:       28420078        slti    v0,v0,120
  402810:       1040001a        beqz    v0,40287c <decode_file+0xfc>
  402814:       00000000        nop
  402818:       8fc40048        lw      a0,72(s8)
  40281c:       8f998058        lw      t9,-32680(gp)
  402820:       00000000        nop
```

```
402824:       0320f809        jalr    t9
402828:       00000000        nop
40282c:       8fdc0010        lw      gp,16(s8)
402830:       afc20024        sw      v0,36(s8)
402834:       8fc30024        lw      v1,36(s8)
402838:       2402ffff        li      v0,-1
40283c:       14620003        bne     v1,v0,40284c <decode_file+0xcc>
402840:       00000000        nop
402844:       1000000d        b       40287c <decode_file+0xfc>
402848:       00000000        nop
40284c:       8fc30020        lw      v1,32(s8)
402850:       8fc20030        lw      v0,48(s8)
402854:       00000000        nop
402858:       00621821        addu    v1,v1,v0
40285c:       93c20024        lbu     v0,36(s8)
402860:       00000000        nop
402864:       a0620000        sb      v0,0(v1)
402868:       8fc20030        lw      v0,48(s8)
40286c:       00000000        nop
402870:       24420001        addiu   v0,v0,1
402874:       1000ffe3        b       402804 <decode_file+0x84>
402878:       afc20030        sw      v0,48(s8)
40287c:       8fc30024        lw      v1,36(s8)
402880:       2402ffff        li      v0,-1
402884:       14620003        bne     v1,v0,402894 <decode_file+0x114>
402888:       00000000        nop
40288c:       24020001        li      v0,1
402890:       a3c20018        sb      v0,24(s8)
402894:       8fc20030        lw      v0,48(s8)
402898:       00000000        nop
40289c:       10400021        beqz    v0,402924 <decode_file+0x1a4>
4028a0:       00000000        nop
4028a4:       27c20028        addiu   v0,s8,40
4028a8:       8fc40020        lw      a0,32(s8)
4028ac:       8fc50030        lw      a1,48(s8)
4028b0:       00403021        move    a2,v0
4028b4:       8f998074        lw      t9,-32652(gp)
4028b8:       00000000        nop
4028bc:       0320f809        jalr    t9
4028c0:       00000000        nop
4028c4:       8fdc0010        lw      gp,16(s8)
4028c8:       afc2001c        sw      v0,28(s8)
4028cc:       8fc2001c        lw      v0,28(s8)
4028d0:       00000000        nop
4028d4:       04410004        bgez    v0,4028e8 <decode_file+0x168>
4028d8:       00000000        nop
4028dc:       2402ffff        li      v0,-1
```

```
  4028e0:       10000019        b       402948 <decode_file+0x1c8>
  4028e4:       afc20034        sw      v0,52(s8)
  4028e8:       8fc40028        lw      a0,40(s8)
  4028ec:       24050001        li      a1,1
  4028f0:       8fc6001c        lw      a2,28(s8)
  4028f4:       8fc7004c        lw      a3,76(s8)
  4028f8:       8f998048        lw      t9,-32696(gp)
  4028fc:       00000000        nop
  402900:       0320f809        jalr    t9
  402904:       00000000        nop
  402908:       8fdc0010        lw      gp,16(s8)
  40290c:       8fc40028        lw      a0,40(s8)
  402910:       8f99802c        lw      t9,-32724(gp)
  402914:       00000000        nop
  402918:       0320f809        jalr    t9
  40291c:       00000000        nop
  402920:       8fdc0010        lw      gp,16(s8)
  402924:       8fc40020        lw      a0,32(s8)
  402928:       8f99802c        lw      t9,-32724(gp)
  40292c:       00000000        nop
  402930:       0320f809        jalr    t9
  402934:       00000000        nop
  402938:       8fdc0010        lw      gp,16(s8)
  40293c:       1000ff9c        b       4027b0 <decode_file+0x30>
  402940:       00000000        nop
  402944:       afc00034        sw      zero,52(s8)
  402948:       8fc20034        lw      v0,52(s8)
  40294c:       03c0e821        move    sp,s8
  402950:       8fbf0040        lw      ra,64(sp)
  402954:       8fbe003c        lw      s8,60(sp)
  402958:       03e00008        jr      ra
  40295c:       27bd0048        addiu   sp,sp,72

00402960 <main>:
  402960:       3c1c0fc0        lui     gp,0xfc0
  402964:       279c56d0        addiu   gp,gp,22224
  402968:       0399e021        addu    gp,gp,t9
  40296c:       27bdffc0        addiu   sp,sp,-64
  402970:       afbc0010        sw      gp,16(sp)
  402974:       afbf0038        sw      ra,56(sp)
  402978:       afbe0034        sw      s8,52(sp)
  40297c:       afbc0030        sw      gp,48(sp)
  402980:       03a0f021        move    s8,sp
  402984:       afc40040        sw      a0,64(s8)
  402988:       afc50044        sw      a1,68(s8)
  40298c:       24020001        li      v0,1
  402990:       afc20018        sw      v0,24(s8)
```

```
402994:       afc0001c        sw      zero,28(s8)
402998:       8fc30040        lw      v1,64(s8)
40299c:       24020001        li      v0,1
4029a0:       14620021        bne     v1,v0,402a28 <main+0xc8>
4029a4:       00000000        nop
4029a8:       8f848070        lw      a0,-32656(gp)
4029ac:       8f858070        lw      a1,-32656(gp)
4029b0:       00000000        nop
4029b4:       24a50058        addiu   a1,a1,88
4029b8:       8f9980bc        lw      t9,-32580(gp)
4029bc:       00000000        nop
4029c0:       0320f809        jalr    t9
4029c4:       00000000        nop
4029c8:       8fdc0010        lw      gp,16(s8)
4029cc:       afc20020        sw      v0,32(s8)
4029d0:       8fc20020        lw      v0,32(s8)
4029d4:       00000000        nop
4029d8:       1040000f        beqz    v0,402a18 <main+0xb8>
4029dc:       00000000        nop
4029e0:       8f848070        lw      a0,-32656(gp)
4029e4:       00000000        nop
4029e8:       248400b0        addiu   a0,a0,176
4029ec:       8f858018        lw      a1,-32744(gp)
4029f0:       00000000        nop
4029f4:       24a53714        addiu   a1,a1,14100
4029f8:       8f868018        lw      a2,-32744(gp)
4029fc:       00000000        nop
402a00:       24c63718        addiu   a2,a2,14104
402a04:       8f998088        lw      t9,-32632(gp)
402a08:       00000000        nop
402a0c:       0320f809        jalr    t9
402a10:       00000000        nop
402a14:       8fdc0010        lw      gp,16(s8)
402a18:       8fc20020        lw      v0,32(s8)
402a1c:       00000000        nop
402a20:       10000226        b       4032bc <main+0x95c>
402a24:       afc2002c        sw      v0,44(s8)
402a28:       8fc30040        lw      v1,64(s8)
402a2c:       24020002        li      v0,2
402a30:       14620069        bne     v1,v0,402bd8 <main+0x278>
402a34:       00000000        nop
402a38:       8fc20018        lw      v0,24(s8)
402a3c:       00000000        nop
402a40:       00021880        sll     v1,v0,0x2
402a44:       8fc20044        lw      v0,68(s8)
402a48:       00000000        nop
402a4c:       00621021        addu    v0,v1,v0
```

```
402a50:       8c440000        lw      a0,0(v0)
402a54:       8f858018        lw      a1,-32744(gp)
402a58:       00000000        nop
402a5c:       24a53750        addiu   a1,a1,14160
402a60:       8f998060        lw      t9,-32672(gp)
402a64:       00000000        nop
402a68:       0320f809        jalr    t9
402a6c:       00000000        nop
402a70:       8fdc0010        lw      gp,16(s8)
402a74:       14400008        bnez    v0,402a98 <main+0x138>
402a78:       00000000        nop
402a7c:       8f9980b0        lw      t9,-32592(gp)
402a80:       00000000        nop
402a84:       0320f809        jalr    t9
402a88:       00000000        nop
402a8c:       8fdc0010        lw      gp,16(s8)
402a90:       1000004e        b       402bcc <main+0x26c>
402a94:       00000000        nop
402a98:       8fc20018        lw      v0,24(s8)
402a9c:       00000000        nop
402aa0:       00021880        sll     v1,v0,0x2
402aa4:       8fc20044        lw      v0,68(s8)
402aa8:       00000000        nop
402aac:       00621021        addu    v0,v1,v0
402ab0:       8c440000        lw      a0,0(v0)
402ab4:       8f858018        lw      a1,-32744(gp)
402ab8:       00000000        nop
402abc:       24a53754        addiu   a1,a1,14164
402ac0:       8f998060        lw      t9,-32672(gp)
402ac4:       00000000        nop
402ac8:       0320f809        jalr    t9
402acc:       00000000        nop
402ad0:       8fdc0010        lw      gp,16(s8)
402ad4:       1440000e        bnez    v0,402b10 <main+0x1b0>
402ad8:       00000000        nop
402adc:       8f848018        lw      a0,-32744(gp)
402ae0:       00000000        nop
402ae4:       24843714        addiu   a0,a0,14100
402ae8:       8f858018        lw      a1,-32744(gp)
402aec:       00000000        nop
402af0:       24a53758        addiu   a1,a1,14168
402af4:       8f9980b8        lw      t9,-32584(gp)
402af8:       00000000        nop
402afc:       0320f809        jalr    t9
402b00:       00000000        nop
402b04:       8fdc0010        lw      gp,16(s8)
402b08:       10000030        b       402bcc <main+0x26c>
```

```
402b0c:       00000000        nop
402b10:       8fc20018        lw      v0,24(s8)
402b14:       00000000        nop
402b18:       00021880        sll     v1,v0,0x2
402b1c:       8fc20044        lw      v0,68(s8)
402b20:       00000000        nop
402b24:       00621021        addu    v0,v1,v0
402b28:       8c440000        lw      a0,0(v0)
402b2c:       8f858018        lw      a1,-32744(gp)
402b30:       00000000        nop
402b34:       24a5375c        addiu   a1,a1,14172
402b38:       8f998060        lw      t9,-32672(gp)
402b3c:       00000000        nop
402b40:       0320f809        jalr    t9
402b44:       00000000        nop
402b48:       8fdc0010        lw      gp,16(s8)
402b4c:       14400011        bnez    v0,402b94 <main+0x234>
402b50:       00000000        nop
402b54:       8f848070        lw      a0,-32656(gp)
402b58:       00000000        nop
402b5c:       248400b0        addiu   a0,a0,176
402b60:       8f858018        lw      a1,-32744(gp)
402b64:       00000000        nop
402b68:       24a53714        addiu   a1,a1,14100
402b6c:       8f868018        lw      a2,-32744(gp)
402b70:       00000000        nop
402b74:       24c63760        addiu   a2,a2,14176
402b78:       8f998088        lw      t9,-32632(gp)
402b7c:       00000000        nop
402b80:       0320f809        jalr    t9
402b84:       00000000        nop
402b88:       8fdc0010        lw      gp,16(s8)
402b8c:       1000000f        b       402bcc <main+0x26c>
402b90:       00000000        nop
402b94:       8f848070        lw      a0,-32656(gp)
402b98:       00000000        nop
402b9c:       248400b0        addiu   a0,a0,176
402ba0:       8f858018        lw      a1,-32744(gp)
402ba4:       00000000        nop
402ba8:       24a53714        addiu   a1,a1,14100
402bac:       8f868018        lw      a2,-32744(gp)
402bb0:       00000000        nop
402bb4:       24c63798        addiu   a2,a2,14232
402bb8:       8f998088        lw      t9,-32632(gp)
402bbc:       00000000        nop
402bc0:       0320f809        jalr    t9
402bc4:       00000000        nop
```

```
402bc8:       8fdc0010        lw      gp,16(s8)
402bcc:       2402ffff        li      v0,-1
402bd0:       100001ba        b       4032bc <main+0x95c>
402bd4:       afc2002c        sw      v0,44(s8)
402bd8:       8fc20040        lw      v0,64(s8)
402bdc:       00000000        nop
402be0:       28420003        slti    v0,v0,3
402be4:       144001a3        bnez    v0,403274 <main+0x914>
402be8:       00000000        nop
402bec:       8fc20040        lw      v0,64(s8)
402bf0:       00000000        nop
402bf4:       28420008        slti    v0,v0,8
402bf8:       1040019e        beqz    v0,403274 <main+0x914>
402bfc:       00000000        nop
402c00:       8fc20018        lw      v0,24(s8)
402c04:       00000000        nop
402c08:       00021880        sll     v1,v0,0x2
402c0c:       8fc20044        lw      v0,68(s8)
402c10:       00000000        nop
402c14:       00621021        addu    v0,v1,v0
402c18:       8c440000        lw      a0,0(v0)
402c1c:       8f858018        lw      a1,-32744(gp)
402c20:       00000000        nop
402c24:       24a5375c        addiu   a1,a1,14172
402c28:       8f998060        lw      t9,-32672(gp)
402c2c:       00000000        nop
402c30:       0320f809        jalr    t9
402c34:       00000000        nop
402c38:       8fdc0010        lw      gp,16(s8)
402c3c:       1440019e        bnez    v0,4032b8 <main+0x958>
402c40:       00000000        nop
402c44:       8fc20018        lw      v0,24(s8)
402c48:       00000000        nop
402c4c:       24420001        addiu   v0,v0,1
402c50:       afc20018        sw      v0,24(s8)
402c54:       8fc20018        lw      v0,24(s8)
402c58:       00000000        nop
402c5c:       00021880        sll     v1,v0,0x2
402c60:       8fc20044        lw      v0,68(s8)
402c64:       00000000        nop
402c68:       00621021        addu    v0,v1,v0
402c6c:       8c440000        lw      a0,0(v0)
402c70:       8f858018        lw      a1,-32744(gp)
402c74:       00000000        nop
402c78:       24a537cc        addiu   a1,a1,14284
402c7c:       8f998060        lw      t9,-32672(gp)
402c80:       00000000        nop
```

```
402c84:       0320f809        jalr    t9
402c88:       00000000        nop
402c8c:       8fdc0010        lw      gp,16(s8)
402c90:       14400004        bnez    v0,402ca4 <main+0x344>
402c94:       00000000        nop
402c98:       24020001        li      v0,1
402c9c:       10000025        b       402d34 <main+0x3d4>
402ca0:       a3c20024        sb      v0,36(s8)
402ca4:       8fc20018        lw      v0,24(s8)
402ca8:       00000000        nop
402cac:       00021880        sll     v1,v0,0x2
402cb0:       8fc20044        lw      v0,68(s8)
402cb4:       00000000        nop
402cb8:       00621021        addu    v0,v1,v0
402cbc:       8c440000        lw      a0,0(v0)
402cc0:       8f858018        lw      a1,-32744(gp)
402cc4:       00000000        nop
402cc8:       24a537d4        addiu   a1,a1,14292
402ccc:       8f998060        lw      t9,-32672(gp)
402cd0:       00000000        nop
402cd4:       0320f809        jalr    t9
402cd8:       00000000        nop
402cdc:       8fdc0010        lw      gp,16(s8)
402ce0:       14400003        bnez    v0,402cf0 <main+0x390>
402ce4:       00000000        nop
402ce8:       10000012        b       402d34 <main+0x3d4>
402cec:       a3c00024        sb      zero,36(s8)
402cf0:       8f848070        lw      a0,-32656(gp)
402cf4:       00000000        nop
402cf8:       248400b0        addiu   a0,a0,176
402cfc:       8f858018        lw      a1,-32744(gp)
402d00:       00000000        nop
402d04:       24a53714        addiu   a1,a1,14100
402d08:       8f868018        lw      a2,-32744(gp)
402d0c:       00000000        nop
402d10:       24c637dc        addiu   a2,a2,14300
402d14:       8f998088        lw      t9,-32632(gp)
402d18:       00000000        nop
402d1c:       0320f809        jalr    t9
402d20:       00000000        nop
402d24:       8fdc0010        lw      gp,16(s8)
402d28:       2402ffff        li      v0,-1
402d2c:       10000163        b       4032bc <main+0x95c>
402d30:       afc2002c        sw      v0,44(s8)
402d34:       8fc20018        lw      v0,24(s8)
402d38:       00000000        nop
402d3c:       24420001        addiu   v0,v0,1
```

```
402d40:       afc20018        sw      v0,24(s8)
402d44:       8fc20018        lw      v0,24(s8)
402d48:       00000000        nop
402d4c:       00021880        sll     v1,v0,0x2
402d50:       8fc20044        lw      v0,68(s8)
402d54:       00000000        nop
402d58:       00621021        addu    v0,v1,v0
402d5c:       8c420000        lw      v0,0(v0)
402d60:       00000000        nop
402d64:       14400030        bnez    v0,402e28 <main+0x4c8>
402d68:       00000000        nop
402d6c:       93c20024        lbu     v0,36(s8)
402d70:       00000000        nop
402d74:       1040000c        beqz    v0,402da8 <main+0x448>
402d78:       00000000        nop
402d7c:       8f848070        lw      a0,-32656(gp)
402d80:       8f858070        lw      a1,-32656(gp)
402d84:       00000000        nop
402d88:       24a50058        addiu   a1,a1,88
402d8c:       8f9980bc        lw      t9,-32580(gp)
402d90:       00000000        nop
402d94:       0320f809        jalr    t9
402d98:       00000000        nop
402d9c:       8fdc0010        lw      gp,16(s8)
402da0:       1000000b        b       402dd0 <main+0x470>
402da4:       afc2001c        sw      v0,28(s8)
402da8:       8f848070        lw      a0,-32656(gp)
402dac:       8f858070        lw      a1,-32656(gp)
402db0:       00000000        nop
402db4:       24a50058        addiu   a1,a1,88
402db8:       8f9980c8        lw      t9,-32568(gp)
402dbc:       00000000        nop
402dc0:       0320f809        jalr    t9
402dc4:       00000000        nop
402dc8:       8fdc0010        lw      gp,16(s8)
402dcc:       afc2001c        sw      v0,28(s8)
402dd0:       8fc2001c        lw      v0,28(s8)
402dd4:       00000000        nop
402dd8:       1040000f        beqz    v0,402e18 <main+0x4b8>
402ddc:       00000000        nop
402de0:       8f848070        lw      a0,-32656(gp)
402de4:       00000000        nop
402de8:       248400b0        addiu   a0,a0,176
402dec:       8f858018        lw      a1,-32744(gp)
402df0:       00000000        nop
402df4:       24a53714        addiu   a1,a1,14100
402df8:       8f868018        lw      a2,-32744(gp)
```

```
402dfc:       00000000        nop
402e00:       24c63718        addiu   a2,a2,14104
402e04:       8f998088        lw      t9,-32632(gp)
402e08:       00000000        nop
402e0c:       0320f809        jalr    t9
402e10:       00000000        nop
402e14:       8fdc0010        lw      gp,16(s8)
402e18:       8fc2001c        lw      v0,28(s8)
402e1c:       00000000        nop
402e20:       10000126        b       4032bc <main+0x95c>
402e24:       afc2002c        sw      v0,44(s8)
402e28:       8fc20018        lw      v0,24(s8)
402e2c:       00000000        nop
402e30:       00021880        sll     v1,v0,0x2
402e34:       8fc20044        lw      v0,68(s8)
402e38:       00000000        nop
402e3c:       00621021        addu    v0,v1,v0
402e40:       8c440000        lw      a0,0(v0)
402e44:       8f858018        lw      a1,-32744(gp)
402e48:       00000000        nop
402e4c:       24a53820        addiu   a1,a1,14368
402e50:       8f998060        lw      t9,-32672(gp)
402e54:       00000000        nop
402e58:       0320f809        jalr    t9
402e5c:       00000000        nop
402e60:       8fdc0010        lw      gp,16(s8)
402e64:       144000f2        bnez    v0,403230 <main+0x8d0>
402e68:       00000000        nop
402e6c:       8fc20018        lw      v0,24(s8)
402e70:       00000000        nop
402e74:       24420001        addiu   v0,v0,1
402e78:       afc20018        sw      v0,24(s8)
402e7c:       8fc20018        lw      v0,24(s8)
402e80:       00000000        nop
402e84:       00021880        sll     v1,v0,0x2
402e88:       8fc20044        lw      v0,68(s8)
402e8c:       00000000        nop
402e90:       00621021        addu    v0,v1,v0
402e94:       8c420000        lw      v0,0(v0)
402e98:       00000000        nop
402e9c:       14400012        bnez    v0,402ee8 <main+0x588>
402ea0:       00000000        nop
402ea4:       8f848070        lw      a0,-32656(gp)
402ea8:       00000000        nop
402eac:       248400b0        addiu   a0,a0,176
402eb0:       8f858018        lw      a1,-32744(gp)
402eb4:       00000000        nop
```

```
402eb8:       24a53714        addiu   a1,a1,14100
402ebc:       8f868018        lw      a2,-32744(gp)
402ec0:       00000000        nop
402ec4:       24c63824        addiu   a2,a2,14372
402ec8:       8f998088        lw      t9,-32632(gp)
402ecc:       00000000        nop
402ed0:       0320f809        jalr    t9
402ed4:       00000000        nop
402ed8:       8fdc0010        lw      gp,16(s8)
402edc:       2402ffff        li      v0,-1
402ee0:       100000f6        b       4032bc <main+0x95c>
402ee4:       afc2002c        sw      v0,44(s8)
402ee8:       8fc20018        lw      v0,24(s8)
402eec:       00000000        nop
402ef0:       00021880        sll     v1,v0,0x2
402ef4:       8fc20044        lw      v0,68(s8)
402ef8:       00000000        nop
402efc:       00621021        addu    v0,v1,v0
402f00:       8c440000        lw      a0,0(v0)
402f04:       8f858018        lw      a1,-32744(gp)
402f08:       00000000        nop
402f0c:       24a53844        addiu   a1,a1,14404
402f10:       8f99806c        lw      t9,-32660(gp)
402f14:       00000000        nop
402f18:       0320f809        jalr    t9
402f1c:       00000000        nop
402f20:       8fdc0010        lw      gp,16(s8)
402f24:       afc20020        sw      v0,32(s8)
402f28:       8fc20020        lw      v0,32(s8)
402f2c:       00000000        nop
402f30:       14400012        bnez    v0,402f7c <main+0x61c>
402f34:       00000000        nop
402f38:       8f848070        lw      a0,-32656(gp)
402f3c:       00000000        nop
402f40:       248400b0        addiu   a0,a0,176
402f44:       8f858018        lw      a1,-32744(gp)
402f48:       00000000        nop
402f4c:       24a53714        addiu   a1,a1,14100
402f50:       8f868018        lw      a2,-32744(gp)
402f54:       00000000        nop
402f58:       24c63848        addiu   a2,a2,14408
402f5c:       8f998088        lw      t9,-32632(gp)
402f60:       00000000        nop
402f64:       0320f809        jalr    t9
402f68:       00000000        nop
402f6c:       8fdc0010        lw      gp,16(s8)
402f70:       2402ffff        li      v0,-1
```

```
402f74:       100000d1        b       4032bc <main+0x95c>
402f78:       afc2002c        sw      v0,44(s8)
402f7c:       8fc20018        lw      v0,24(s8)
402f80:       00000000        nop
402f84:       24420001        addiu   v0,v0,1
402f88:       afc20018        sw      v0,24(s8)
402f8c:       8fc20018        lw      v0,24(s8)
402f90:       00000000        nop
402f94:       00021880        sll     v1,v0,0x2
402f98:       8fc20044        lw      v0,68(s8)
402f9c:       00000000        nop
402fa0:       00621021        addu    v0,v1,v0
402fa4:       8c420000        lw      v0,0(v0)
402fa8:       00000000        nop
402fac:       14400018        bnez    v0,403010 <main+0x6b0>
402fb0:       00000000        nop
402fb4:       8f848070        lw      a0,-32656(gp)
402fb8:       00000000        nop
402fbc:       248400b0        addiu   a0,a0,176
402fc0:       8f858018        lw      a1,-32744(gp)
402fc4:       00000000        nop
402fc8:       24a53714        addiu   a1,a1,14100
402fcc:       8f868018        lw      a2,-32744(gp)
402fd0:       00000000        nop
402fd4:       24c63888        addiu   a2,a2,14472
402fd8:       8f998088        lw      t9,-32632(gp)
402fdc:       00000000        nop
402fe0:       0320f809        jalr    t9
402fe4:       00000000        nop
402fe8:       8fdc0010        lw      gp,16(s8)
402fec:       8fc40020        lw      a0,32(s8)
402ff0:       8f998064        lw      t9,-32668(gp)
402ff4:       00000000        nop
402ff8:       0320f809        jalr    t9
402ffc:       00000000        nop
403000:       8fdc0010        lw      gp,16(s8)
403004:       2402ffff        li      v0,-1
403008:       100000ac        b       4032bc <main+0x95c>
40300c:       afc2002c        sw      v0,44(s8)
403010:       8fc20018        lw      v0,24(s8)
403014:       00000000        nop
403018:       00021880        sll     v1,v0,0x2
40301c:       8fc20044        lw      v0,68(s8)
403020:       00000000        nop
403024:       00621021        addu    v0,v1,v0
403028:       8c440000        lw      a0,0(v0)
40302c:       8f858018        lw      a1,-32744(gp)
```

```
403030:       00000000        nop
403034:       24a538a8        addiu   a1,a1,14504
403038:       8f998060        lw      t9,-32672(gp)
40303c:       00000000        nop
403040:       0320f809        jalr    t9
403044:       00000000        nop
403048:       8fdc0010        lw      gp,16(s8)
40304c:       14400061        bnez    v0,4031d4 <main+0x874>
403050:       00000000        nop
403054:       8fc20018        lw      v0,24(s8)
403058:       00000000        nop
40305c:       24420001        addiu   v0,v0,1
403060:       afc20018        sw      v0,24(s8)
403064:       8fc20018        lw      v0,24(s8)
403068:       00000000        nop
40306c:       00021880        sll     v1,v0,0x2
403070:       8fc20044        lw      v0,68(s8)
403074:       00000000        nop
403078:       00621021        addu    v0,v1,v0
40307c:       8c420000        lw      v0,0(v0)
403080:       00000000        nop
403084:       14400018        bnez    v0,4030e8 <main+0x788>
403088:       00000000        nop
40308c:       8f848070        lw      a0,-32656(gp)
403090:       00000000        nop
403094:       248400b0        addiu   a0,a0,176
403098:       8f858018        lw      a1,-32744(gp)
40309c:       00000000        nop
4030a0:       24a53714        addiu   a1,a1,14100
4030a4:       8f868018        lw      a2,-32744(gp)
4030a8:       00000000        nop
4030ac:       24c63888        addiu   a2,a2,14472
4030b0:       8f998088        lw      t9,-32632(gp)
4030b4:       00000000        nop
4030b8:       0320f809        jalr    t9
4030bc:       00000000        nop
4030c0:       8fdc0010        lw      gp,16(s8)
4030c4:       8fc40020        lw      a0,32(s8)
4030c8:       8f998064        lw      t9,-32668(gp)
4030cc:       00000000        nop
4030d0:       0320f809        jalr    t9
4030d4:       00000000        nop
4030d8:       8fdc0010        lw      gp,16(s8)
4030dc:       2402ffff        li      v0,-1
4030e0:       10000076        b       4032bc <main+0x95c>
4030e4:       afc2002c        sw      v0,44(s8)
4030e8:       8fc20018        lw      v0,24(s8)
```

```
4030ec:       00000000        nop
4030f0:       00021880        sll     v1,v0,0x2
4030f4:       8fc20044        lw      v0,68(s8)
4030f8:       00000000        nop
4030fc:       00621021        addu    v0,v1,v0
403100:       8c440000        lw      a0,0(v0)
403104:       8f858018        lw      a1,-32744(gp)
403108:       00000000        nop
40310c:       24a538ac        addiu   a1,a1,14508
403110:       8f99806c        lw      t9,-32660(gp)
403114:       00000000        nop
403118:       0320f809        jalr    t9
40311c:       00000000        nop
403120:       8fdc0010        lw      gp,16(s8)
403124:       afc20028        sw      v0,40(s8)
403128:       93c20024        lbu     v0,36(s8)
40312c:       00000000        nop
403130:       1040000a        beqz    v0,40315c <main+0x7fc>
403134:       00000000        nop
403138:       8fc40020        lw      a0,32(s8)
40313c:       8fc50028        lw      a1,40(s8)
403140:       8f9980bc        lw      t9,-32580(gp)
403144:       00000000        nop
403148:       0320f809        jalr    t9
40314c:       00000000        nop
403150:       8fdc0010        lw      gp,16(s8)
403154:       10000009        b       40317c <main+0x81c>
403158:       afc2001c        sw      v0,28(s8)
40315c:       8fc40020        lw      a0,32(s8)
403160:       8fc50028        lw      a1,40(s8)
403164:       8f9980c8        lw      t9,-32568(gp)
403168:       00000000        nop
40316c:       0320f809        jalr    t9
403170:       00000000        nop
403174:       8fdc0010        lw      gp,16(s8)
403178:       afc2001c        sw      v0,28(s8)
40317c:       8fc2001c        lw      v0,28(s8)
403180:       00000000        nop
403184:       1040000f        beqz    v0,4031c4 <main+0x864>
403188:       00000000        nop
```