

# Linux

- . Framebuffer
  - Framebuffer
- -
- -
- -
- -

# . Framebuffer

# Framebuffer

## . Framebuffer

### 1.1 LCD Framebuffer

LCD Framebuffer	LCD Framebuffer	LCD	480000	LCD
4=960000 1.92MB	1.92MB			

FramebufferAPP\_Image00001

Image not found or type unknown

Framebuffer	LCD	Framebuffer
-------------	-----	-------------

### 1.2 Framebuffer API

#### 1.2.1 open

FramebufferAPP\_Image00002

Image not found or type unknown

```
#include <sys/types.h> #include <sys/stat.h> #include <fcntl.h>
```

- int open(const char \*pathname, int flags);
- int open(const char \*pathname, int flags, mode\_t mode);

- pathname
- Flags

- ① O\_RDWR ;
- ② O\_RDONLY ;
- ③ O\_WRONLY ;
- ④ O\_APPEND ;
- ⑤ O\_TRUNC
- ⑥ O\_CREAT O\_EXCL

Mode flags O\_CREAT

-1

## 1.2.2 ioctl

FramebufferAPP\_Image00003

Image not found or type unknown

```
#include <sys/ioctl.h>
```

- int ioctl(int fd, unsigned long request, ...);

- fd
- request
- ... arg request

-1

## 1.2.3 mmap

FramebufferAPP\_Image00004

Image not found or type unknown

```
#include <sys/mman.h>
```

- void \*mmap(void \*addr, size\_t length, int prot, int flags,int fd, off\_t offset);

- addr NULL

- length
  - prot 4
    - ①PROT\_EXEC
    - ②PROT\_READ
    - ③PROT\_WRITE
    - ④PROT\_NONE
  - Flags
    - ①MAP\_SHARED
    - ②MAP\_PRIVATE
- 1

## 1.3 LCD

### 1.3.1 LCD

FramebufferAPP\_Image00005

Image not found or type unknown

‘A’ ‘1’ ‘0’ 8\*16

### 1.3.2 fb\_var\_screeninfo

fb0 LCD fb\_info

FramebufferAPP\_Image00006

Image not found or type unknown

ioctl xres(x) yres y bits\_per\_pixel

show\_ascii.c

```

4718     fd_fb = open("/dev/fb0", O_RDWR);
4719     if ( fd_fb < 0)
4720     {
4721         printf("can't open /dev/fb0\n");
4722         return -1;
4723     }
4724     if (ioctl(fd_fb, FBIOGET_VSCREENINFO, &var))
  
```

```

4725     {
4726     printf("can't get var\n");
4727     return -1;
4728 }

```

LCD fb0 ioctl fb\_var\_screeninfo var var xres(x) yre

## 1.3.3 fb\_var\_screeninfo

fb\_var\_screeninfo var var

xres bits\_per\_pixel

show\_ascii.c

```

4730 line_width = var.xres * var.bits_per_pixel / 8;

```

bits\_per\_pixel

show\_ascii.c

```

4731 pixel_width = var.bits_per_pixel / 8;

```

xres yres bits\_per\_pixel

show\_ascii.c

```

4732 screen_size = var.xres * var.yres * var.bits_per_pixel / 8;

```

## 1.3.4 mmap

show\_ascii.c

```

4733 fbmem = (unsigned char *)mmap(NULL, screen_size, PROT_READ | PROT_WRITE, MAP_SHARED,
4734 fd_fb, 0);
4735 if (fbmem == (unsigned char *)-1)
4736 {
4737     printf("can't mmap\n");
4738     return -1;
4739 }
4740 /* : */

```

```
4741memset( fbmem, 0, screen_size);
```

mmap

PROT\_READ | PROT\_WRITE

MAP\_SHARED

fb

## 1.3.5

show\_ascii.c

```
4641void lcd_put_pixel(int x, int y, unsigned int color)
```

3 x y

show\_ascii.c

```
4643unsigned char *pen_8 = fbmem+y*line_width+x*pixel_width;
4644unsigned short *pen_16;
4645unsigned int *pen_32;
4646
4647unsigned int red, green, blue;
4648
4649pen_16 = (unsigned short *)pen_8;
4650pen_32 = (unsigned int *)pen_8;
```

x y

1

2

4

fbmem

mmap

fbmem

1

= fbmem+Y \*

+ x \*

show\_ascii.c

```
4652switch ( var.bits_per_pixel)
4653{
4654case 8:
4655{
4656pen_8 = color;
4657break;
4658}
4659case 16:
4660{
4661pen_16/* 565 */
```

```

4662    red    = (color >> 16) & 0xff;
4663    green  = (color >> 8) & 0xff;
4664    blue   = (color >> 0) & 0xff;
4665    color  = ((red >> 3) << 11) | ((green >> 2) << 5) | (blue >> 3);
4666    *pen_16 = color;
4667    break;
4668}
4669case 32:
4670{
4671    *pen_32 = color;
4672    break;
4673}
4674default:
4675{
4676    printf("can't support %dbpp\n", var.bits_per_pixel);
4677    break;
4678}
4679}
4680}

```

fb0    bits\_per\_pixel    pen   pen\_8   pen\_16   pen\_32    color    pen

## 1.4 LCD

### 1.4.1 LCD

①                    c                    ASCII

show\_ascii.c

```

4693    unsigned char *dots = (unsigned char *)&fontdata_8x16[c*16];

```

②                    '1'                    '0'

FramebufferAPP\_Image00007

Image not found or type unknown

LCD

16

8

8



show\_ascii.c

```
4697 for (i = 0; i < 16; i++)
4698 {
4699     byte = dots[i];
4700     for (b = 7; b >= 0; b--)
4701     {
4702         if (byte & (1<<b))
4703         {
4704             /* show */
4705             lcd_put_pixel(x+7-b, y+i, 0xffffffff); /* */
4706         }
4707     } else
4708     {
4709         /* hide */
4710         lcd_put_pixel(x+7-b, y+i, 0); /* */
4711     }
4712 }
4713 }
```

③ lcd\_put\_ascii

show\_ascii.c

```
4743 lcd_put_ascii(var.xres/2, var.yres/2, 'A'); /* 8*16 A*/
```

④ c show\_ascii.c

arm-linux-gnueabi-hf-gcc -o show\_ascii show\_ascii.c

⑤ show\_ascii show\_ascii

./show\_ascii

'A'

## 1.4.2 LCD

FramebufferAPP\_Image00008

Image not found or type unknown

(HZK16)

①

show\_font.c

```
4760 fd_hzk16 = open("HZK16", O_RDONLY);
```

②

hzk\_stat

show\_font.c

```
4793 if(fstat(fd_hzk16, &hzk_stat))
```

mmap

③

mmap

show\_font.c

```
4798 hzkmem = (unsigned char *)mmap(NULL , hzk_stat.st_size, PROT_READ, MAP_SHARED, fd_hzk16
0);
```

hzkmem fbmem

④

HZK16 GB2312 16×16 HZK16 32 ‘ ’ D6D0 2

show\_font.c

```
4734 unsigned int area = str[0] - 0xA1;
4735 unsigned int where = str[1] - 0xA1;
4736 unsigned char *dots = hzkmem + (area * 94 + where)*32;
```

FramebufferAPP\_Image00009

Image not found or type unknown

16

2

‘1’

show\_font.c

```
4740 for (i = 0; i < 16; i++)
4741 for (j = 0; j < 2; j++)
```

### 1.5.1 freetype

## ① freetype

```
tar xjf freetype-2.4.10.tar.bz2
```

## ② freetype-2.4.10

```
cd freetype-2.4.10
```

## ③ freetype-2.4.10

```
./configure --host=arm-linux-gnueabi --prefix=/home/book/100ask_imx6ull-sdk/ToolChain/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi/arm-linux-gnueabi/libc/usr/
```

## ④ internal

```
mkdir /home/book/100ask_imx6ull-sdk/ToolChain/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi/arm-linux-gnueabi/libc/usr/include/freetype2/freetype/internal -p
```

## ④

```
make
```

## ⑤

```
make install
```

## ⑥ freetype

```
mv /home/book/100ask_imx6ull-sdk/ToolChain/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi/arm-linux-gnueabi/libc/usr/include/freetype2/freetype /home/book/100ask_imx6ull-sdk/ToolChain/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabi/arm-linux-gnueabi/libc/usr/include/
```

# 1.5.2 freetype

100ask

freetype

freetype

/home/book/100ask\_imx6ull-sdk/ToolChain/gcc-linaro-6.2.1-2016.11-x86\_64\_arm-linux-gnueabi/arm-linux-gnueabi/libc/usr/include/\*

/home/book/100ask\_imx6ull-sdk/ToolChain/gcc-linaro-6.2.1-2016.11-x86\_64\_arm-linux-gnueabi/arm-linux-gnueabi/libc/usr/lib/so

# 1.6 freetype

## 1.5.1

① A

FramebufferAPP\_Image00010

Image not found or type unknown

②

FramebufferAPP\_Image00011

Image not found or type unknown

③ A

FramebufferAPP\_Image00012

Image not found or type unknown

1/2 2/4

## 1.5.2 Freetype

Freetype	API	freetype
glyph	Windows	FONTS
TTF	simsun.ttc	
FramebufferAPP_Image00013		
Image not found or type unknown		
Charmaps	GBK UNICODE BIG5	charmap
		glyph

- ① 'A' 0x41 ' ' GBK UNICODE ,BIG5
- ② charmap glyph
- ③
- ④ glyph
- ⑤
- ⑥ LCD

FramebufferAPP\_Image00014

Image not found or type unknown

step1 step2 step3 freetype

- ① FT\_InitFreetype
- ② Face FT\_New\_Face
- ③ FT\_Set\_Char\_Sizes FT\_Set\_Pixel\_Sizes
- ④ charmap FT\_Select\_Charmap
- ⑤ charcode glyph : glyph\_index = FT\_Get\_Char\_Index face charcode
- ⑥ glyph\_index glyph FT\_Load\_Glyph face glyph\_index
- ⑦ FT\_Render\_Glyph
- ⑧ :FT\_Set\_Transform

## 1.5.2 LCD

FramebufferAPP\_Image00015

Image not found or type unknown

C

- ① freetype

freetype\_show\_font.c

```
4872 error = FT_Init_FreeType( &library );  
/* initialize library */
```

- ② freetype FT\_New\_Face face &face

freetype\_show\_font.c

```
4875     error = FT_New_Face( library, argv[1], 0, &face ); /* create face object */
```

③ face glyph

freetype\_show\_font.c

```
4877     slot = face->glyph;
```

④ 24\*24

freetype\_show\_font.c

```
4879     FT_Set_Pixel_Sizes( face, 24, 0);
```

⑤

LCD x y freetype x y

FramebufferAPP\_Image00017

Image not found or type unknown

‘ ‘ lcd ‘ ‘

x y ‘A’ xres/2 yres/2

- lcd\_x = var.xres/2 + 8 + 16 lcd\_y = var.yres/2 + 16
- :x = lcd\_x = var.xres/2 + 8 + 16 y = var.yres - lcd\_y = var.yres/2 - 16
- 1/64 64

freetype\_show\_font.c

```
4888    pen.x = ( var.xres/2 + 8 + 16) * 64;
4889    pen.y = ( var.yres/2 - 16) * 64;
4890
4891    /* set transformation */
4892    FT_Set_Transform( face, 0, &pen);
```

⑥ glyph

```
4895     error = FT_Load_Char( face, chinese_str[0], FT_LOAD_RENDER );
4896     if ( error)
4897     {
```

```

4898    printf("FT_Load_Char error\n");
4899    return -1;
4900}

```

## FramebufferAPP\_Image00018

Image not found or type unknown

FT\_Load\_Char 3

example1.c

freetype\_show\_font.c

```

4902  draw_bitmap( &slot->bitmap,
4903              slot->bitmap_left,
4904              var.yres - slot->bitmap_top);

```

example1.c

## FramebufferAPP\_Image00019

Image not found or type unknown

3

- Width LCD x var.xres
- Height LCD y var.yres
- lcd\_put\_pixel image  
lcd\_put\_pixel(i, j, bitmap->buffer[q \* bitmap->width + p]);

⑥ C freetype\_show\_font.c

arm-linux-gnueabi-gcc -finput-charset=GBK -fexec-charset=GBK -o freetype\_show\_font  
freetype\_show\_font.c -lfreetype -lm

⑦ freetype\_show\_font simsun.ttc simsun.ttc freetype\_show\_font

./freetype\_show\_font ../simsun.ttc

‘ ‘

## 1.5.3 LCD



## example1.c

### FramebufferAPP\_Image00020

Image not found or type unknown

### freetype\_show\_font\_angle.c

```
4888 /* use 25 degrees */
4894 angle = ( 1.0 * strtoul( argv[ 2], NULL, 0) / 360 ) * 3.14159 * 2;

4895 /* set up matrix */
4896 matrix.xx = (FT_Fixed)( cos( angle ) * 0x10000L );
4897 matrix.xy = (FT_Fixed)( -sin( angle ) * 0x10000L );
4898 matrix.yx = (FT_Fixed)( sin( angle ) * 0x10000L );
4899 matrix.yy = (FT_Fixed)( cos( angle ) * 0x10000L );
4900
4901 /* set transformation */
4902 FT_Set_Transform( face, &matrix, &pen);
```

```
arm-linux-gnueabi-hf-gcc -finput-charset=GBK -fexec-charset=GBK -o freetype_show_font_angle
freetype_show_font_angle.c -lfreetype -lm
```

freetype\_show\_font\_angle

freetype\_show\_font

simsun.ttc

```
./freetype_show_font_angle ../simsun.ttc 90
```

‘ ‘ 90



# 2

## 2.1 BMP

### 2.1.1 BMP

BMP                      BMP                      4                      (bitmap-file header)                      (bitmap-information header)                      (color table)

1.                      (bitmap-file header)

bfType	2	BMP 0x42,0x4D ASCII    "B""M"
bfSize	4	BMP
bfReserved1	2	
bfReserved2	2	

UltraEdit                      BMP

ImageProcess\_Image001

Image not found or type unknown

BMP                      32                      0x42,0x4D

4                      0x36,0xF9,0x15,0x00

BMP

0x36,0xF9,0x15,0x00                      0x0015F936(                      )

ImageProcess\_Image002

Image not found or type unknown

4                      0x00

4                      +                      +                      0x36

2.                      (bitmap-information header)

biSize	4	
biWidth	4	
biHeight	4	
biPlanes	2	1
biBitCount	2	1 1 4 8 16 24 32                      24
biCompression	4	
biSizeImage	4	
biXPelsPerMeter	4	/
biYPelsPerMeter	4	/
biClrUsed	4	
biClrImportant	4	0

ImageProcess\_Image003

Image not found or type unknown

0E-11 00000028h = 40,                      40

12-15 00000320h = 800800

16-19 00000258h = 600600X Y

1A-1B 0001h,1

1C-1D 0018h = 24,2424

ImageProcess\_Image004

Image not found or type unknown

1E-21 00000000h BI\_RGB

22-25 00000000hBI\_RGB0

26-29 00000000h

2A-2D 00000000h

2E-31 00000000h240

32-35 00000000h240

3. (color palette)

24

4.

24 BGR

# 2.1.2 BMP RGB LCD

BMP LCD

1. LCD RGB2 RGB5653 RGB88824 BMP BG

2. LCD LCD BMP

2.1 24 BMP RGB

2.1

1. ✓\*\*\*\*\*

2. ✗ \* IsBmp

```

3.  *          BMP
4.  *          ptFileMap -
5.  *
6.  *          0 - BMP , -1 - BMP
7.  * *****/
8.  int IsBmp(FILE **ppFp, const char *strFileName)
9.  {
10.     char strCheckHeader[2];
11.     *ppFp= fopen(strFileName, "rb+");
12.     if (*ppFp== NULL) {
13.         return -1;
14.     }
15.     if (fread(strCheckHeader, 1, 2, *ppFp) != 2)
16.         return -1;
17.
18.     if (strCheckHeader[0] != 0x42 || strCheckHeader[1] != 0x4d)
19.         return -1;
20.     else
21.         return 0;
22. }
23.
24.
25.
26. /******/
27.  *          MapFile
28.  *          mmap          ,
29.  *          PT_PictureData ptData
30.  *          ptData->iFileSize :
31.  *          ptData->pucFileData :
32.  *          0 - -
33.  * *****/
34.  int MapFile(PT_PictureData ptData)
35.  {
36.     int iFd;
37.     struct stat tStat;
38.
39.     /* */
40.     iFd = fileno(ptData->ptFp);
41.     fstat(iFd, &tStat);
42.     ptData->iFileSize= tStat.st_size;

```

```

43. □   pData->pucFileData= (unsigned char *)mmap(NULL , tStat.st_size, PROT_READ |
PROT_WRITE, MAP_SHARED, iFd, 0);
44. □   if (pData->pucFileData == (unsigned char *)-1)
45. □   {
46. □       printf("mmap error!\n");
47. □       return -1;
48. □   }
49. □   return 0;
50. □}
51. □
52. □/*****
53. □ *      DecodeBmp2Rgb
54. □ *      BMP      rgb
55. □ *      strFileName -
56. □ *                  pData -
57. □ *      0      -      -
58. □ *                  -1      -      BMP
59. □ *                  -2      -      bpp
60. □ *                  -3      -
61. □ *****/
62. □static int DecodeBmp2Rgb(const char *strFileName, PT_PictureData pData) {
63. □   int x,y;
64. □   int iPos = 0;
65. □   int iLineWidthAlign;
66. □   BITMAPFILEHEADER *ptBITMAPFILEHEADER;
67. □   BITMAPINFOHEADER *ptBITMAPINFOHEADER;
68. □   unsigned char *aFileHead;
69. □   unsigned char *pucSrc;
70. □   unsigned char *pucDest;
71. □   int iLineBytes;
72. □
73. □   /*      BMP      */
74. □   if (IsBmp(&pData->ptFp, strFileName))
75. □       return -1;
76. □
77. □   /* BMP      */
78. □   MapFile(pData);
79. □
80. □
81. □   aFileHead = pData->pucFileData;

```

```

82. □
83. □   ptBITMAPFILEHEADER = (BITMAPFILEHEADER *)aFileHead;
84. □   ptBITMAPINFOHEADER = (BITMAPINFOHEADER *)(aFileHead + sizeof(BITMAPFILEHEADER));
85. □   /*           */
86. □   pData->iWidth = ptBITMAPINFOHEADER->biWidth;
87. □   pData->iHeight = ptBITMAPINFOHEADER->biHeight;
88. □   pData->iBpp = ptBITMAPINFOHEADER->biBitCount;
89. □       iLineBytes = pData->iWidth*pData->iBpp/8; //
90. □   pData->iBmpDataSize= pData->iHeight * iLineBytes; // BMP
91. □   /*    24bpp    */
92. □   if (pData->iBpp != 24)
93. □   {
94. □       printf("iBMPBpp = %d\n", pData->iBpp);
95. □       printf("sizeof(BITMAPFILEHEADER) = %d\n", sizeof(BITMAPFILEHEADER));
96. □       return -2;
97. □   }
98. □
99. □   /*           */
100. □   pData->pucBmpData = malloc(pData->iBmpDataSize);
101. □   pData->pucRgbData = malloc(pData->iBmpDataSize);
102. □
103. □   if (NULL == pData->pucBmpData || NULL == pData->pucRgbData)
104. □       return -2;
105. □
106. □   /* bmp          24bpp BMP   BGR   */
107. □   pucDest = pData->pucBmpData;
108. □   iLineWidthAlign = (iLineBytes + 3) & ~0x3;   /* 4   */
109. □   pucSrc = aFileHead + ptBITMAPFILEHEADER->bfOffBits;
110. □
111. □   pucSrc = pucSrc + (pData->iHeight - 1) * iLineWidthAlign;
112. □
113. □   /*   bmp                               */
114. □   for (y = 0; y < pData->iHeight; y++)
115. □   {
116. □       memcpy(pucDest, pucSrc, pData->iWidth*3);
117. □       pucSrc -= iLineWidthAlign;
118. □       pucDest += iLineBytes;
119. □   }
120. □
121. □

```



```

122. □    /*    BGR    RGB    */
123. □    for ( y = 0; y < pData->iHeight; y++){
124. □        for( x = 0; x<pData->iWidth*3; x+=3){
125. □            pData->pucRgbData[ iPos++] = pData->pucBmpData[ y*pData->iWidth*3+x+2];
126. □            pData->pucRgbData[ iPos++] = pData->pucBmpData[ y*pData->iWidth*3+x+1];
127. □            pData->pucRgbData[ iPos++] = pData->pucBmpData[ y*pData->iWidth*3+x+0];
128. □        }
129. □    }
130. □
131. □    return 0;
132. □
133. □}

```

## 2.2 JPEG

### 2.2.1 JPEG libjpeg

JPEG .jpg      JPEG BMP JPEG      BMP      JPEG JPEG  
 BMP JPEG      Linux jpeg  
 libjpeg jpeg      libjpeg X86 ARM libjpeg  
 libjpeg libjpeg libjpeg

1.

```

tar xzf libjpeg-turbo-1.2.1.tar.gz
cd libjpeg-turbo-1.2.1/

```

2.

```

tar xzf libjpeg-turbo-1.2.1.tar.gz
./configure --prefix=/work/projects/libjpeg-turbo-1.2.1/tmp/ --host=arm-linux
make
make install

```

3.

```
cd /work/projects/libjpeg-turbo-1.2.1/tmp/include
cp * /usr/local/arm/4.3.2/arm-none-linux-gnueabi/libc/usr/include
cd /work/projects/libjpeg-turbo-1.2.1/tmp/lib
cp *.so* -d /usr/local/arm/4.3.2/arm-none-linux-gnueabi/libc/armv4t/lib
```

```
cd /work/projects/libjpeg-turbo-1.2.1/tmp/lib
cp *.so* /work/nfs_root/fs_mini_mdev_new/lib/ -d
```

### 2.2.2 libjpeg

```
libjpeg          libjpeg.txt  example.c libjpeg
```

### 1. jpeg\_compress\_struct

```
cinfo.err = jpeg_std_error(&jerr);  
jpeg_create_decompress(&cinfo);
```

```
jpeg_stdio_src(&cinfo, infile);
```

```
1 1 jpeg_compress_struct
```

2 JPEG

3. jpg

```
jpeg_read_header(&cinfo, TRUE);
```

cinfo	image_width	image_height	cinfo	scale_num	scale_den
-------	-------------	--------------	-------	-----------	-----------

```
jpeg_start_decompress( &cinfo);
```

cinfo cinfo

```
jpeg_read_scanlines(&cinfo, buffer, 1);
```

6.

```
jpeg_finish_decompress(&cinfo);
```

7. jpeg\_compress\_struct

```
jpeg_destroy_decompress(&cinfo);
```

## 2.2.3 libjpeg JPEG RGB LCD

JPEG RGB

2.2

```
1. /******
2.  *      IsJpg
3.  *      Jpg
4.  *      ptData -
5.  *      strFileName -
6.  *      0 - JPEG - JPEG
7.  * *****/
8. static int IsJpg(PictureData ptData, const char *strFileName)
9. {
10.     int iRet;
11.
12.     jpeg_stdio_src(&ptData->tInfo, ptData->ptFp);
13.
14.     /* jpeg_read_header jpeg */
15.     iRet = jpeg_read_header(&ptData->tInfo, TRUE);
16.
17.     return (iRet == JPEG_HEADER_OK);
18. }
19.
20. /******
21.  *      DecodeJpg2Rgb
22.  *      JPG      RGB888
23.  *      ptData -
```

```

24. □ *                strFileName -
25. □ *    PT_PictureData->pucRgbData -    rgb
26. □ *    0 -    -
27. □ *****/
28. □ static int DecodeJpg2Rgb(const char *strFileName, PT_PictureData ptData){
29. □     int iRowSize;
30. □     unsigned char *pucbuffer;
31. □     unsigned char *pucHelp; //
32. □
33. □     /* 1.         jpeg_compress_struct     */
34. □     ptData->tInfo.err = jpeg_std_error(&ptData->tJerr);
35. □     jpeg_create_decompress(&ptData->tInfo);
36. □
37. □
38. □     /* 2.         */
39. □     if ((ptData->ptFp= fopen(strFileName, "rb")) == NULL) {
40. □         fprintf(stderr, "can't open %s\n", strFileName);
41. □         return -1;
42. □     }
43. □
44. □     /* 3.    jpg                JPEG     */
45. □     if (!IsJpg(ptData, strFileName)) {
46. □         printf("file is not jpg ... \n");
47. □         return -1;
48. □     }
49. □
50. □
51. □
52. □     /*         */
53. □     ptData->tInfo.scale_num = 1;
54. □     ptData->tInfo.scale_denom = 1;
55. □     /* 4.         jpeg_start_decompress */
56. □     jpeg_start_decompress(&ptData->tInfo);
57. □
58. □
59. □     /*         tInfo         */
60. □     ptData->iWidth= ptData->tInfo.output_width;
61. □     ptData->iHeight = ptData->tInfo.output_height;
62. □     ptData->iBpp = ptData->tInfo.output_components*8;
63. □     /*         */

```

```

64. □   iRowSize = pData->iWidth * pData->tInfo.output_components;
65. □   pucbuffer = malloc( iRowSize);
66. □   pData->iRgbSize= iRowSize * pData->iHeight;
67. □   pData->pucRgbData = malloc(pData->iRgbSize);
68. □
69. □   /* pucHelp  pData->pucRgbData   */
70. □   pucHelp = pData->pucRgbData;
71. □   /* 5.   jpeg_read_scanlines           */
72. □   while ( pData->tInfo.output_scanline < pData->tInfo.output_height)
73. □   {
74. □       /*   jpeg_read_scanlines       pucbuffer   */
75. □       jpeg_read_scanlines(&pData->tInfo, &pucbuffer, 1);
76. □       /*           */
77. □       memcpy( pucHelp, pucbuffer, iRowSize);
78. □       pucHelp += iRowSize;
79. □   }
80. □   free(pucbuffer);
81. □   /* 6.   */
82. □   jpeg_finish_decompress(&pData->tInfo);
83. □   /* 7.   jpeg_compress_struct   */
84. □   jpeg_destroy_decompress(&pData->tInfo);
85. □   return 0;
86. □ }

```

## 2.3 PNG

### 2.3.1 PNG libpng

JPEG

PNG

JPEG

PNG

LZ77

PNG

libpng

libpng

<http://www.libpng.org/pub/png/libpng.html>

libpng

libpng

libpng

1.

```
tar xzf libpng-1.6.37.tar.gz
cd libpng-1.6.37/
```

2.

```
./configure --prefix=/work/projects/libpng-1.6.37/tmp/ --host=arm-linux
make
make install
```

3.

```
cd /work/projects/libpng-1.6.37/tmp/include
cp * /usr/local/arm/4.3.2/arm-none-linux-gnueabi/libc/usr/include
cd /work/projects/libpng-1.6.37/tmp/lib
cp *.so* -d /usr/local/arm/4.3.2/arm-none-linux-gnueabi/libc/armv4t/lib
```

4.

```
cd /work/projects/libpng-1.6.37/tmp/lib
cp *.so* /work/nfs_root/fs_mini_mdev_new/lib/ -d
```

## 2.3.2 libpng

libpng                      libpng-manual.txt    example.c   libjpeg

1.              libpng              png\_ptr info\_ptr  
     A. png\_ptr = png\_create\_read\_struct(PNG\_LIBPNG\_VER\_STRING, NULL, NULL, NULL);  
     2 3 4                              NULL

B. info\_ptr = png\_create\_info\_struct(png\_ptr);

2.  
     setjmp(png\_jmpbuf(png\_ptr));  
                          libpng                              png\_create\_read\_struct

3.  
     png\_init\_io(png\_ptr, fp);  
     1 1      png\_ptr      2      PNG

4.   PNG  
     A.  
     png\_read\_png(png\_ptr, info\_ptr, png\_transforms, png\_voidp\_NULL);  
                          info\_ptr                              png\_transforms                              libpng  
     B.

png\_get\_image\_width png\_get\_image\_height png\_get\_color\_type      png

5. info\_ptr

PNG

A.

png\_read\_image(png\_ptr, row\_pointers);

1 1 png\_ptr 2

B.

row\_pointers = png\_get\_rows(png\_ptr, info\_ptr);

1 2 1 png\_ptr, info\_ptr

1 1 png\_ptr 2

6.

png\_destroy\_read\_struct(&png\_ptr, &info\_ptr, 0);

## 2.3.3 libpng png rgb LCD

2.3

```
1. /******
2.  *      IsNotPng
3.  *      PNG
4.  *      ppFp -
5.  *      strFileName -
6.  *      0 - PNG - PNG
7.  * *****/
8. int IsNotPng(FILE **ppFp, const char *strFileName)
9. {
10.     char strCheckHeader[8];
11.     *ppFp= fopen(strFileName, "rb");
12.     if (*ppFp== NULL) {
13.         return -1;
14.     }
15.     /* PNG 8 png_sig_cmp PNG */
16.     if (fread(strCheckHeader, 1, 8, *ppFp) != 8)
17.         return -1;
18.     return png_sig_cmp(strCheckHeader, 0, 8);
19.
20. }
21.
22. /******
23.  *      DecodePng2Rgb
24.  *      PNG      RGB888
```

```

25. □ *      pData -
26. □ *
27. □ *      PT_PictureData->pucRgbData -   rgb
28. □ *      0 -      -
29. □ *****/
30. □ static int DecodePng2Rgb(const char *strFileName, PT_PictureData pData)
31. □ {
32. □     int i, j;
33. □     int iPos = 0;
34. □     png_bytepp pucPngData;
35. □     /* 0.      PNG */
36. □     if (IsnotPng(&pData->ptFp, strFileName)) {
37. □         printf("file is not png ... \n");
38. □         return -1;
39. □     }
40. □
41. □     /* 1.      libpng      png_ptr info_ptr */
42. □     pData->ptPngStrPoint = png_create_read_struct(PNG_LIBPNG_VER_STRING, NULL, NULL,
NULL);
43. □     pData->ptPngInfoPoint= png_create_info_struct(pData->ptPngStrPoint);
44. □
45. □     /* 2.      */
46. □     setjmp( png_jmpbuf( pData->ptPngStrPoint ));
47. □     rewind(pData->ptFp); // fseek( fp, 0, SEEK_SET);
48. □
49. □     /* 3.      */
50. □     png_init_io(pData->ptPngStrPoint, pData->ptFp);
51. □
52. □     /* 4.  PNG
53. □         *   PNG_TRANSFORM_EXPAND
54. □         *   PNG      BGR888 ABGR8888 */
55. □     png_read_png(pData->ptPngStrPoint, pData->ptPngInfoPoint, PNG_TRANSFORM_EXPAND,
0);
56. □     pData->iChannels      = png_get_channels(pData->ptPngStrPoint, pData-
>ptPngInfoPoint);
57. □     pData->iWidth        = png_get_image_width(pData->ptPngStrPoint, pData-
>ptPngInfoPoint);
58. □     pData->iHeight       = png_get_image_height(pData->ptPngStrPoint, pData-
>ptPngInfoPoint);
59. □

```



```

60. □
61. □  /* 5. info_ptr          */
62. □  pucPngData = png_get_rows(ptData->ptPngStrPoint, ptData->ptPngInfoPoint);
//      png_get_rowbytes();
63. □  if (ptData->iChannels == 4) { //      24      32
64. □      ptData->iRawSize= ptData->iWidth * ptData->iHeight*4; //
65. □      ptData->pucRawData= (unsigned char*)malloc(ptData->iRawSize);
66. □      if (NULL == ptData->pucRawData) {
67. □          printf("malloc rgba faile ...\n");
68. □          png_destroy_read_struct(&ptData->ptPngStrPoint, &ptData->ptPngInfoPoint, 0);
69. □          fclose(ptData->ptFp);
70. □          return -1;
71. □      }
72. □      /* pucPngData      RGBA
73. □      *      ABGR */
74. □      for (i = 0; i < ptData->iHeight; i++)
75. □          for (j = 0; j < ptData->iWidth * 4; j += 4) {
76. □              ptData->pucRawData[iPos++] = pucPngData[i][j + 3];
77. □              ptData->pucRawData[iPos++] = pucPngData[i][j + 2];
78. □              ptData->pucRawData[iPos++] = pucPngData[i][j + 1];
79. □              ptData->pucRawData[iPos++] = pucPngData[i][j + 0];
80. □          }
81. □
82. □      /*      RGBA      RGB888      */
83. □      if(RgbaToRgb( ptData)!=0)
84. □          return -1;
85. □
86. □  }
87. □  else if (ptData->iChannels == 3 ) { //      24      32
88. □      ptData->iRgbSize= ptData->iWidth * ptData->iHeight*3; //
89. □      ptData->pucRgbData = (unsigned char*)malloc(ptData->iRgbSize);
90. □      if (NULL == ptData->pucRgbData) {
91. □          printf("malloc rgba faile ...\n");
92. □          png_destroy_read_struct(&ptData->ptPngStrPoint, &ptData->ptPngInfoPoint, 0);
93. □          fclose(ptData->ptFp);
94. □          return -1;
95. □      }
96. □      /* pucPngData      RGB
97. □      *      BGR */
98. □      for (i = 0; i < ptData->iHeight; i ++ ) {

```

```

99.     for ( j = 0; j < ptData->iWidth*3; j += 3) {
100.         pData->pucRgbData[ iPos++] = pucPngData[ i][ j+2];
101.         pData->pucRgbData[ iPos++] = pucPngData[ i][ j+1];
102.         pData->pucRgbData[ iPos++] = pucPngData[ i][ j+0];
103.     }
104. }
105.     pData->iBpp = 24; //      RGB888
106. }
107. else return -1;
108.
109.
110. /* 6: */
111. png_destroy_read_struct(&ptData->ptPngStrPoint, &ptData->ptPngInfoPoint, 0);
112. fclose( ptData->ptFp);
113.
114.
115. return 0;
116. }

```

## 2.4

### 2.4.1

#### 2.4.1.1

"lantianyu520" " "

200 100

400 200

(0,0)

(0,0),(0,100),(200,0),(200,100)

(40,50) x

40/200=0.2 y

50/100=0.5

Dx,Dy

Dx/400

(Sx,Sy)

(Dx,Dy)

Sw,Sh

Dw,Dh

$$S_x/D_x = S_w/D_w \quad S_y/D_y = S_h/D_h$$

$$S_x = D_x * S_w/D_w \quad S_y = D_y * S_h/D_h$$

## 2.4.1.2

### 2.4

```

1.  /******
2.  *      PicZoom
3.  *
4.  *      ,      free
5.  *      "      "      "lantianyu520"      "      "
6.  *      ptPicData -
7.  *      fSize -
8.  *      ptPicData->pucZoomData,
9.  *      0 - , -
10. /* *****/
11. int PicZoom(PT_PictureData ptPicData, float fSize)
12. {
13.     ptPicData->iZoomWidth = ptPicData->iWidth * fSize;
14.     ptPicData->iZoomHeight= ptPicData->iHeight* fSize;
15.     unsigned long* pdwSrcXTable;
16.     unsigned long x;
17.     unsigned long y;
18.     unsigned long dwSrcY;
19.     unsigned char *pucDest;
20.     unsigned char *pucSrc;
21.     unsigned long dwPixelBytes = ptPicData->iBpp/8;
22.     ptPicData->pucZoomData= malloc(sizeof(unsigned char) * ptPicData->iZoomWidth*ptPicData-
    >iZoomHeight*ptPicData->iBpp/8);
23.     pdwSrcXTable = malloc(sizeof(unsigned long) * ptPicData->iZoomWidth);
24.     if (NULL == pdwSrcXTable){
25.         printf("malloc error!\n");
26.         return -1;
27.     }
28.
29.     /*      for      Sx = Dx * Sw/Dw Sy = Dy * Sh/Dh*/
30.     for (x = 0; x < ptPicData->iZoomWidth; x++){//      pdwSrcXTable

```

```

31. □      /*      for      x
32. □      * pdwSrcXTable[ x]      Sx,
33. □      * x      Dx,
34. □      * ptPicData->iWidth      Sw
35. □      * ptPicData->iZoomWidth      Dw*/
36. □      pdwSrcXTable[ x]=( x*ptPicData->iWidth/ptPicData->iZoomWidth);
37. □  }
38. □
39. □  for (y = 0; y < ptPicData->iZoomHeight; y++){
40. □  /*      2      y
41. □      * dwSrcY      Sy,
42. □      * y      Dy,
43. □      * ptPicData->iHeight      Sh
44. □      * ptPicData->iZoomHeight      Dh*/
45. □      dwSrcY = (y * ptPicData->iHeight / ptPicData->iZoomHeight);
46. □  /*      RGB      */
47. □      pucDest = ptPicData->pucZoomData + y*ptPicData->iZoomWidth*3;
48. □      pucSrc = ptPicData->pucRgbData + dwSrcY*ptPicData->iWidth*3;
49. □
50. □  /*      */
51. □      for (x = 0; x <ptPicData->iZoomWidth; x++){
52. □          memcpy(pucDest+x*dwPixelBytes, pucSrc+pdwSrcXTable[ x]*dwPixelBytes,
dwPixelBytes);
53. □      }
54. □  }
55. □
56. □  free(pdwSrcXTable);
57. □  return 0;
58. □

```

## 2.4.2

### 2.4.2.1

“ “ “ “

A

B

A

B

B

B

x0 y0

r

x

b

a

x1 y1 ,

ImageProcess\_Image005

Image not found or type unknown

x0=rcosb y0=rsinb

 $x1 = r\cos(b-a) = r\cos b\cos a + r\sin b\sin a = x0\cos a + y0\sin a$  $y1 = r\sin(b-a) = r\sin b\cos a - r\cos b\sin a = -x0\sin a + y0\cos a$ 

RGB

A

B x1,y1

A

## 2.4.2.2

2.5

```

1. #define PI 3.1415926535
2. //
3. #define RADIAN(angle) ((angle)*PI/180.0)
4.
5.
6.
7.
8.
9. typedef struct ConcernCoor {
10.     int iLTx; // left top x
11.     int iLTy; //left top y
12.     int iLBx; //left bottom x
13.     int iLBy; //left bottom y
14.     int iRTx; //right top x
15.     int iRTy; //right top y
16.     int iRBx; // right bottom x
17.     int iRBy; // right bottom y
18. } T_ConcernCoor, *PT_ConcernCoor;

```

```

19. □
20. □
21. □/*****
22. □ *      max
23. □ *
24. □ *      x y  int
25. □ *
26. □ *      x y
27. □ *****/
28. □ static int max(int x,int y){
29. □     return x>y?x:y;
30. □ }
31. □/*****
32. □ *      PicRotate
33. □ *
34. □ *      ,      free
35. □ *      "      "      "      "
36. □ *      ptPicData -
37. □ *      fAngle      -      0<=angle<=360
38. □ *      ptPicData->pucRotateData,      rgb
39. □ *      0 -      ,      -
40. □ *****/
41. □ int PicRotate(PT_PictureData ptPicData,float fAngle)
42. □ {
43. □     int i ,j;
44. □     T_ConcernCoor tConCor,tRonCor;
45. □     //
46. □     //int iSrcLineSize = bitCount * srcW / 8;
47. □     int iSrcLineSize = ptPicData->iBpp* ptPicData->iZoomWidth / 8;
48. □     int iDesLineSize;
49. □     int iX; //      x
50. □     int iY; //      y
51. □
52. □     /*      A      B      ,
53. □     *
54. □     * tConCor      B
55. □     * tRonCor      B      */
56. □     tConCor.iLTx = -ptPicData->iZoomWidth/2; tConCor.iLTy = ptPicData->iZoomHeight/2;
57. □     tConCor.iRTx = ptPicData->iZoomWidth/2; tConCor.iRTy = ptPicData->iZoomHeight/2;
58. □     tConCor.iLBx = -ptPicData->iZoomWidth/2; tConCor.iLBy = -ptPicData->iZoomHeight/2;
59. □     tConCor.iRBx = ptPicData->iZoomWidth/2; tConCor.iRBy = ptPicData->iZoomHeight/2;

```

```

60. □
61. □
62. □    /*      B      */
63. □    double sina = sin( RADIAN( fAngle ));
64. □    double cosa = cos( RADIAN( fAngle ));
65. □    tRonCor.iLTx =tConCor.iLTx * cosa + tConCor.iLTy * sina;
66. □    tRonCor.iLTy = -tConCor.iLTx * sina + tConCor.iLTy * cosa;
67. □    tRonCor.iRTx =tConCor.iRTx * cosa + tConCor.iRTy * sina;
68. □    tRonCor.iRTy = -tConCor.iRTx * sina + tConCor.iRTy * cosa;
69. □    tRonCor.iLBx = tConCor.iLBx * cosa + tConCor.iLBy * sina;
70. □    tRonCor.iLBy = -tConCor.iLBx * sina + tConCor.iLBy * cosa;
71. □    tRonCor.iRBx = tConCor.iRBx * cosa + tConCor.iRBy * sina;
72. □    tRonCor.iRBy = -tConCor.iRBx * sina + tConCor.iRBy * cosa;
73. □
74. □
75. □    /*          */
76. □    ptPicData->iRotateWidth = max( abs( tRonCor.iRBx - tRonCor.iLTx ), abs( tRonCor.iRTx -
tRonCor.iLBx ));
77. □    ptPicData->iRotateHeight = max( abs( tRonCor.iRBy - tRonCor.iLTy ), abs( tRonCor.iRTy -
tRonCor.iLBy ));
78. □
79. □    /*      3      */
80. □    iDesLineSize = ((ptPicData->iRotateWidth* ptPicData->iBpp+ 23) / 24) * 3 ;
81. □    /*          */
82. □    ptPicData->pucRotateData = malloc( iDesLineSize * ptPicData->iRotateHeight);
83. □    if( NULL == ptPicData->pucRotateData){
84. □        printf("malloc error\n");
85. □        return -1;
86. □    }
87. □
88. □    /*          *
89. □    * i, j      B x1, y1*/
90. □    for ( i = 0; i < ptPicData->iRotateHeight; i++){
91. □        for ( j = 0; j < ptPicData->iRotateWidth; j++){
92. □            /*      B x, y1      iX, iY,      x0, y0 */
93. □            iX = (j - ptPicData->iRotateWidth / 2)*cos( RADIAN( 360 - fAngle)) + (-i +
ptPicData->iRotateHeight / 2)*sin( RADIAN( 360 - fAngle));
94. □            iY = -(j - ptPicData->iRotateWidth / 2)*sin( RADIAN( 360 - fAngle)) + (-i +
ptPicData->iRotateHeight / 2)*cos( RADIAN( 360 - fAngle));
95. □            /*          */
96. □            if ( iX > ptPicData->iZoomWidth / 2 || iX < -ptPicData->iZoomWidth / 2 || iY >

```

```

ptPicData->iZoomHeight / 2 || iY < -ptPicData->iZoomHeight / 2){
97.     continue;
98. }
99.     /*      B  x0,y0      A      */
100.     int iXN = iX + ptPicData->iZoomWidth / 2;
101.     int iYN = abs(iY - ptPicData->iZoomHeight / 2);
102.     /*      */
103.     memcpy( &ptPicData->pucRotateData[i * iDesLineSize + j * 3], &ptPicData-
>pucZoomData[iYN * iSrcLineSize + iXN * 3], 3);
104. }
105. }
106. return 0;
107. }

```





# 3

## 3.1

Linux Linux

## 3.2

Linux input core( ) drivers( ) event handlers( ) Linu

InputSystem\_Image002

Image not found or type unknown

--> --> --> --->

## 3.3

```
cat /proc/bus/input/devices
```

event ubuntu

InputSystem\_Image003

Image not found or type unknown

## I N P S U H B

### I:id of the device( ID)

struct input\_id

```
41 struct input_id {
42     //
43     __u16 bustype;
44     //      ID
45     __u16 vendor;
46     //      ID
47     __u16 product;
48     //      ID
49     __u16 version;
50 };
```

### N:name of the device

### P:physical path to the device in the system hierarchy

### S:sysfs path

sys

### U:unique identification code for the device(if device has it)

### H:list of input handles associated with the device.

### B:bitmaps( )

PROP:device properties and quirks.

EV:types of events supported by the device.

KEY:keys/buttons this device has.

MSC:miscellaneous events supported by the device.

LED: leds present on the device.

PROP:

EV:

KEY: /

MSC:

LED:

```
cat /proc/bus/input/devices
```

event1

event1

**cat**

InputSystem\_Image004

Image not found or type unknown

```
cat /dev/input/event1
```

**hexdump**

InputSystem\_Image005

Image not found or type unknown

input\_event

/usr/include/linux/input.h

input\_event

```
24 struct input_event {
25     /*
26      * struct timeval time;
27      */
28     __u16 type;
29     /*
30      * __u16 code;
31      */
```

```
32 __s32 value;
33 };
```

input\_event      time

```
1 struct timeval
2 {
3     __time_t tv_sec;          /* Seconds. */
4     __suseconds_t tv_usec;    /*Microseconds. */
5 };
```

tv\_sec Epoch      struct timeval      tv\_usec      Epoch      1970

input\_event      type

type      code      X   Y      value

(type)

```
/usr/include/linux/input-event-codes.h
```

Linux

```
/usr/include/linux/input.h
```

```
34 /*
35  * Event types
36  */
37
38 #define EV_SYN 0x00//
39 #define EV_KEY 0x01//
40 #define EV_REL 0x02//
41 #define EV_ABS 0x03//
42 #define EV_MSC 0x04
43 #define EV_SW 0x05
44 #define EV_LED 0x11
45 #define EV_SND 0x12
46 #define EV_REP 0x14
47 #define EV_FF 0x15
48 #define EV_PWR 0x16
49 #define EV_FF_STATUS 0x17
```

```
50 #define EV_MAX(0xf)
51 #define EV_CNT( EV_MAX+1)
```

**(code)**

:

```
/usr/include/linux/input-event-codes.h
```

Linux

```
/usr/include/linux/input.h
64 /*
65  * Keys and buttons
66  *
67  * Most of the keys/buttons are modeled after USB HUT 1.12
68  * (see http://www.usb.org/developers/hidpage).
69  * Abbreviations in the comments:
70  * AC - Application Control
71  * AL - Application Launch Button
72  * SC - System Control
73  */
74
75 #define KEY_RESERVED 0
76 #define KEY_ESC 1
77 #define KEY_1 2
78 #define KEY_2 3
79 #define KEY_3 4
80 #define KEY_4 5
81 #define KEY_5 6
82 #define KEY_6 7
83 #define KEY_7 8
84 #define KEY_8 9
85 #define KEY_9 10
86 #define KEY_0 11
87 #define KEY_MINUS 12
88 #define KEY_EQUAL 13
89 #define KEY_BACKSPACE 14
90 #define KEY_TAB 15
91 #define KEY_Q 16
```

```
92 #define KEY_W[0]L7
...

```

(value)

# 3.4 USB

USB cat /proc/bus/input/devices USB ev

InputSystem\_Image006

Image not found or type unknown

hexdump :

InputSystem\_Image007

Image not found or type unknown

3.3 USB

1 (type)

3.3

```
/usr/include/linux/input-event-codes.h

```

Linux

```
/usr/include/linux/input.h

```

```
34 /*
35  * Event types
36  */
37
38 #define EV_SYN[0]x00//
39 #define EV_KEY[0]x01//

```

```

40 #define EV_REL 0x02 //
41 #define EV_ABS 0x03 //
42 #define EV_MSC 0x04
43 #define EV_SW 0x05
44 #define EV_LED 0x11
45 #define EV_SND 0x12
46 #define EV_REP 0x14
47 #define EV_FF 0x15
48 #define EV_PWR 0x16
49 #define EV_FF_STATUS 0x17
50 #define EV_MAX 0x1f
51 #define EV_CNT (EV_MAX+1)

```

## 2 (code)

USB code,

```
/usr/include/linux/input-event-codes.h
```

Linux

```
/usr/include/linux/input.h
```

```

696 /*
697  * Relative axes
698  */
699
700 #define REL_X 0x00 // X
701 #define REL_Y 0x01 // Y
702 #define REL_Z 0x02
703 #define REL_RX 0x03
704 #define REL_RY 0x04
705 #define REL_RZ 0x05
706 #define REL_HWHEEL 0x06
707 #define REL_DIAL 0x07
708 #define REL_WHEEL 0x08
709 #define REL_MISC 0x09
710 #define REL_MAX 0x0f
711 #define REL_CNT (REL_MAX+1)

```



REL\_X REL\_Y

value, (type) (code) X Y

input

```
#include <linux/input.h>
```

1 input\_event input

```
struct input_event event_mouse ;
```

2 input USB event2

```
open("/dev/input/event2", O_RDONLY);
```

3

```
read(fd , &event_mouse , sizeof(event_mouse));
```

4

```
//  
if(EV_ABS == event_mouse.type || EV_REL == event_mouse.type)  
{  
    //code    X Y    X    X    value  
    //    Y    Y    value  
    if(event_mouse.code == REL_X)  
    {  
        printf("event_mouse.code_X: %d\n", event_mouse.code);  
        printf("event_mouse.value_X: %d\n", event_mouse.value);  
    }  
    else if(event_mouse.code == REL_Y)  
    {  
        printf("event_mouse.code_Y: %d\n", event_mouse.code);  
        printf("event_mouse.value_Y: %d\n", event_mouse.value);  
    }  
}
```

5

```
close(fd);
```

## Linux

```
01 #include <stdio.h>
02 #include <unistd.h>
03 #include <stdlib.h>
04 #include <fcntl.h>
05 #include <linux/input.h>
06
07 int main(void)
08 {
09     //1         input
10     struct input_event event_mouse ;
11     //2  input          USB          event2
12     int fd      = -1 ;
13     fd = open("/dev/input/event2", O_RDONLY);
14     if(-1 == fd)
15     {
16         printf("open mouse event fair! \n");
17         return -1 ;
18     }
19     while(1)
20     {
21         //3
22         read(fd, &event_mouse, sizeof(event_mouse));
23         if(EV_ABS == event_mouse.type || EV_REL == event_mouse.type)
24         {
25             //code      X  Y      X      X      value
26             //      Y      Y      value
27             if(event_mouse.code == REL_X)
28             {
29                 printf("event_mouse.code_X: %d\n", event_mouse.code);
30                 printf("event_mouse.value_X: %d\n", event_mouse.value);
31             }
32             else if(event_mouse.code == REL_Y)
33             {
34                 printf("event_mouse.code_Y: %d\n", event_mouse.code);
35                 printf("event_mouse.value_Y: %d\n", event_mouse.value);
```

```

36         }
37     }
38 }
39     close(fd);
40     return 0 ;
41 }

```

```
gcc test_mouse.c -o test_mouse
```

## InputSystem\_Image008

Image not found or type unknown

test\_mouse      test\_mouse

## InputSystem\_Image009

Image not found or type unknown

X                  value X

## InputSystem\_Image010

Image not found or type unknown

Y                  value Y

# 3.5

3.3

3.4

event1,    3.3    3.4

input

```
#include <linux/input.h>
```

1           input\_event    input

```
struct input_event event_keyboard ;
```

2   input                           event1

```
open("/dev/input/event1", O_RDONLY);
```

3

```
read( fd , &event_keyboard , sizeof(event_keyboard));
```

4

```
//
if(EV_KEY == event_keyboard.type)
{
    if(1 == event_keyboard.value)
        printf("      :%d      :%d  \n", event_keyboard.type,       event_keyboard.code);
    else if(0 == event_keyboard.value)
        printf("      :%d      :%d  \n", event_keyboard.type, event_keyboard.code);
}
```

5

```
close(fd);
```

```
01 #include <stdio.h>
02 #include <unistd.h>
03 #include <stdlib.h>
04 #include <fcntl.h>
05 #include <linux/input.h>
06
07 int main(void)
08 {
09     //1           input
10     struct input_event event_keyboard ;
11     //2   input                           event1
12     int fd    = -1 ;
13     fd = open("/dev/input/event1", O_RDONLY);
```

```

14     if(-1 == fd)
15     {
16         printf("open mouse event fair! \n");
17         return -1 ;
18     }
19     while(1)
20     {
21         //3
22         read(fd, &event_keyboard, sizeof(event_keyboard));
23     } if(EV_KEY == event_keyboard.type)
24     {
25         if(1 == event_keyboard.value)
26             printf("      :%d      :%d      \n", event_keyboard.type, event_keyboard.code);
27         else if(0 == event_keyboard.value)
28             printf("      :%d      :%d      \n", event_keyboard.type, event_keyboard.code);
29     }
30     }
31     close(fd);
32     return 0 ;
33 }

```

USB

USB

value

```
gcc test_keyboard.c -o test_keyboard
```

InputSystem\_Image011

Image not found or type unknown

test\_keyboard

test\_keyboard

InputSystem\_Image012

Image not found or type unknown

# 3.6

# imx6ul

input

input X Y

EV\_ABS

X Y

ABS\_MT\_POSITION\_X ABS\_MT\_POSITION\_Y

```
01 #include <stdio.h>
02 #include <unistd.h>
03 #include <fcntl.h>
04 #include <stdlib.h>
05 #include <linux/input.h>
06
07 int main(int argc, char **argv)
08 {
09     int tp_fd = -1 ;
10     int tp_ret = -1 ;
11     int touch_x, touch_y ;
12     struct input_event imx6ull_ts ;
13     //1
14     tp_fd = open("/dev/input/event1", O_RDONLY);
15     if(tp_fd < 0)
16     {
17         printf("open /dev/input/event1 fail!\n");
18         return -1 ;
19     }
20     while(1)
21     {
22         //2
23         read(tp_fd , &imx6ull_ts , sizeof(imx6ull_ts));
24         switch(imx6ull_ts.type)
25         {
26             case EV_ABS:
27                 if(imx6ull_ts.code == ABS_MT_POSITION_X)
28                 {
29                     touch_x = imx6ull_ts.value ;
30                 }
31                 if(imx6ull_ts.code == ABS_MT_POSITION_Y)
32                 {
33                     touch_y = imx6ull_ts.value ;
34                 }
35             }
36     }
```

```

31  while (1) break ;
32  while (1) defalut:
33  while (1) break ;
34  while (1) {
35  while (1) printf("touch_x: %d touch_y: %d\n", touch_x, touch_y);
36  while (1) usleep(100);
37  while (1) }
38  while (1) close(tp_fd);
39  while (1) return 0;
40  }

```

```
gcc test_touchscreen.c -o test_touchscreen
```

( PC )

InputSystem\_Image013

Image not found or type unknown

rz PC test\_touchscreen

InputSystem\_Image014

Image not found or type unknown

11 PC

test\_touchscreen :

InputSystem\_Image015

Image not found or type unknown

test\_touchscreen

InputSystem\_Image016

Image not found or type unknown





# 4 Linux

## 4.1

linux(1) - Linux

### 4.1.1

#### 4.1.1.1

C/C++ C/C++ C/C++

```
01 #include <stdio.h>
02
03 int main(int argc, char *argv[])
04 {
05     printf("hello world! \n");
06     return 0;
07 }
```

#### 4.1.1.2

# 4.1.1.3

1.

2.

3.

4.
- 

# 4.1.1.4


ProcessCommunication\_Image001

Image not found or type unknown

1.

=

=CPU

=

=

2.

=

=

=

=

# 4.1.2

# 4.1.2.1

```
fork
: #include <unistd.h>
: pid_t fork(void);
:      (>0 )      0; fork
: fork
```

: fork

----

1. linux
2. pcb PC PC fork fork
- 3.
- 4.
5. INIT INIT

## Tips

linux API ubuntu man

ProcessCommunication\_Image002

Image not found or type unknown

ProcessCommunication\_Image003

Image not found or type unknown

jz2440\process\1th\_create\_process\create\_process.c

```
01 /*****
02  *
03  *
04  *
05  *
06  *
07  * -----
08  * 2020/05/16      V1.0      zh(ryan)
09  *****/
10
11 #include <stdio.h>
12 #include <stdlib.h>
13 #include <unistd.h>
14 #include <sys/types.h>
15
16 int main(int argc, char *argv[])
17 {
```

```

18     pid_t pid;
19
20     pid = fork();  //
21
22     if (pid == 0) {  //
23         int i = 0;
24         for (i = 0; i < 5; i++) {
25             usleep(100);
26             printf("this is child process i=%d\n", i);
27         }
28     }
29
30     if (pid > 0) {  //
31         int i = 0;
32         for (i = 0; i < 5; i++) {
33             usleep(100);
34             printf("this is parent process i=%d\n", i);
35         }
36     }
37
38     while(1);  //
39     return 0;
40 }

```

## JZ2440

jz2440	NFS	jz2440	ubuntu	NFS	u-t
--------	-----	--------	--------	-----	-----

- 

```
arm-linux-gcc create_process.c -o create_process
```

- test NFS

```
cp create_process /work/nfs_root/first_fs
```

- jz2440

## ProcessCommunication\_Image004

Image not found or type unknown

-

"&"

"&"

```
./create_process &
```

ProcessCommunication\_Image005

Image not found or type unknown

- top

```
top
```

ProcessCommunication\_Image006

Image not found or type unknown

```
create_process      PID 777(      PID 776)      PID 776(      PID 770)
```

## 4.1.2.2

```
exit
: #include <stdlib.h>
: void exit (int status)
```

```
_exit
: #include <unistd.h>
: void _exit(int status);
```

```
exit      _exit
```

```
return      exit _exit      return      main      return
```

```
exit _exit
```

```
1  exit      jz2440\process\2th_exit_process\exit_process.c
```

```
01 /*****
02 *      exit
03 *
04 *
05 *
06 *
07 * -----
08 * 2020/05/16      V1.0      zh(ryan)
```

```

09  *****/
10 #include <stdio.h>
11 #include <stdlib.h>
12
13 int main(int argc, char *argv[])
14 {
15     printf("hello world\n");
16     printf("will exit");
17     exit(0);    // _exit
18 }

```

2 \_exit jz2440\process\3th\_exit\_process\exit\_process.c

```

01 /*****
02 *      _exit
03 *
04 *
05 *
06 *
07 * -----
08 * 2020/05/16      V1.0      zh(ryan)
09 *****/
10 #include <stdio.h>
11 #include <stdlib.h>
12
13 int main(int argc, char *argv[])
14 {
15     printf("hello world\n");
16     printf("will exit");
17     _exit(0);    // _exit
18 }

```

15 16        "\n"

exit \_exit        16        "\n"

## JZ2440

### 1

•

```
arm-linux-gcc exit_process.c -o exit_process
```

- NFS

```
cp exit_process /work/nfs_root/first_fs
```

- 

```
./exit_process
```

“hello world” “will exit”

ProcessCommunication\_Image008

Image not found or type unknown

## 4.1.2.3

```
wait
: #include <sys/types.h>
[] #include <sys/wait.h>
: pid_t wait(int *status);
: -1
```

```
waitpid
: #include <sys/types.h>
[] #include <sys/wait.h>
: pid_t waitpid(pid_t pid, int *status, int options);
: -1
```

jz2440\process\4th\_exit\_wait\exit\_wait.c

```
1 /*****
02 *      exit      waitpid
03 *
04 *
05 *
06 *
07 * -----
08 * 2020/05/16      V1.0      zh(ryan)
09 *****/
10 #include <unistd.h>
11 #include <stdio.h>
12 #include <stdlib.h>
```

```

13 #include <sys/types.h>
14 #include <sys/wait.h>
15
16 int main(int argc, char *argv[])
17 {
18     int status = -1;
19     pid_t pid;
20
21     pid = fork();
22     if (pid == 0){ //
23         printf("fork\n");
24         exit(1);
25     } else if (pid > 0) { //
26         pid = waitpid(pid, &status, 0);
27         printf("status=0x%x\n", status);
28     } else {
29         perror("fork\n");
30     }
31
32     return 0;
33 }

```

## JZ2440

- 

```
arm-linux-gcc exit_wait.c -o exit_wait
```

- NFS

```
cp exit_wait /work/nfs_root/first_fs
```

- 

```
./exit_wait
```

## ProcessCommunication\_Image009

Image not found or type unknown



# 4.2

“global” **Ubuntu**

1

```
01 #include <stdio.h>
02 int global = 1;
03
04 void delay(void)
05 {
06     unsigned int a = 1000000;
07     while(a--);
08 }
09
10 int main(int argc, char *argv[])
11 {
12     while (1) {
13         printf("global=%d\n", global);
14         delay();
15     }
16     return 0;
17 }
```

2

```
01 #include <stdio.h>
02 int global = 2;
03
04 void delay(void)
05 {
06     unsigned int a = 1000000;
07     while(a--);
08 }
09
10 int main(int argc, char *argv[])
11 {
12     while (1) {
13         printf("global=%d\n", global);
14         delay();
```

```
15 }
16 return 0;
17 }
```

ProcessCommunication\_Image010

Image not found or type unknown

- 

```
gcc test1.c -o test1
gcc test2.c -o test2
```

- 

```
./test1
./test2
```

ProcessCommunication\_Image011

Image not found or type unknown

1

ProcessCommunication\_Image012

Image not found or type unknown

2



ProcessCommunication\_Image013

Image not found or type unknown

linux

linux

IPC	
socket	socket      socket

linux

”      ----“

soc

ProcessCommunication\_Image014

Image not found or type unknown

4.3

	(      )

4.3.1

4.3.1.1

fd[0]      fd[1]

open      pipe

ProcessCommunication\_Image015

Image not found or type unknown

4.3.1.2

```

1.  #include <unistd.h>

2.  : int pipe(int fd[2])

3.  :          fd[ 0] fd[ 1]          fd[ 0]   fd[ 1]

4.  0      1

```

### 4.3.1.3

- |    |       |        |        |
|----|-------|--------|--------|
| 1. | read  | fd[ 0] |        |
| 2. | write | fd[ 1] |        |
| 3. | close | fd[ 0] | fd[ 1] |

### 4.3.1.4

#### 1

jz2440\process\_pipe\1th\_write\_pipe\my\_pipe\_write.c

```
01 /*****
02  *
03
04  *
05  *
06  *
07  *
08  * -----
09  * 2020/05/16      V1.0      zh(ryan)
10  *****/
11 #include <stdio.h>
12 #include <stdlib.h>
13 #include <unistd.h>
14
15 int main(int argc, char *argv[])
16 {
17     int fd[2];
18     int ret = 0;
19     char write_buf[] = "Hello linux";
20     char read_buf[128] = {0};
21
22     ret = pipe(fd);
23     if (ret < 0) {
24         printf("create pipe fail\n");
25         return -1;
26     }
27     printf("create pipe sucess fd[ 0]=%d fd[ 1]=%d\n", fd[ 0], fd[ 1]);
```

```

28
29     //     fd[1]
30     write(fd[1], write_buf, sizeof(write_buf));
31
32     //     fd[0]
33     read(fd[0], read_buf, sizeof(read_buf));
34     printf("read_buf=%s\n", read_buf);
35
36     close(fd[0]);
37     close(fd[1]);
38     return 0;
39 }

```

## JZ2440

- 

```
arm-linux-gcc my_pipe_write.c -o my_pipe_write
```

- NFS

```
cp my_pipe_write /work/nfs_root/first_fs
```

- 

```
./my_pipe_write
```

" Hello linux"

## 2

"proce:

jz2440\process\_pipe\2th\_comm\test.c

```

01 /*****
02 *      1.
03      2.      process_inter 1
04      3.      process_inter 1
05 *
06 *
07 *
08 *

```

```

09  * -----
10  * 2020/05/16      V1.0      zh(ryan)
11  *****/
12  #include <stdio.h>
13  #include <stdlib.h>
14  #include <unistd.h>
15  #include <sys/types.h>
16
17  int main(int argc, char *argv[])
18  {
19      pid_t pid;
20      int process_inter = 0;
21
22      pid = fork();  //
23
24      if (pid == 0) {  //
25          int i = 0;
26          while (process_inter == 0); //
27          for (i = 0; i < 5; i++) {
28              usleep(100);
29              printf("this is child process i=%d\n", i);
30          }
31      }
32
33      if (pid > 0) {  //
34          int i = 0;
35          for (i = 0; i < 5; i++) {
36              usleep(100);
37              printf("this is parent process i=%d\n", i);
38          }
39          process_inter == 1;
40      }
41
42      while(1);
43      return 0;
44  }

```

**JZ2440**

•

```
arm-linux-gcc test.c -o test
```

- NFS

```
cp test /work/nfs_root/first_fs
```

- 

```
./test
```

29                    process\_inter    0

ProcessCommunication\_Image017

Image not found or type unknown

**3**

jz2440\process\_pipe\3th\_pipe\_comm\comm\_fork.c

```
01 /*****
02  *      1.
03      2.
04      3.
05  *
06  *
07  *
08  *
09  * -----
10  * 2020/05/16      V1.0      zh(ryan)
11  *****/
12
13 #include <stdio.h>
14 #include <stdlib.h>
15 #include <unistd.h>
16 #include <sys/types.h>
17
18 int main(int argc, char *argv[])
19 {
20     pid_t pid;
21     char process_inter = 0;
22     int fd[2], ret = 0;
```

```

23
24     ret = pipe(fd);    //
25     if (ret < 0) {
26         printf("create pipe fail\n");
27         return -1;
28     }
29     printf("create pipe sucess\n");
30
31     pid = fork();    //
32
33     if (pid == 0) {    //
34         int i = 0;
35         read(fd[0], &process_inter, sizeof(process_inter));    //
36         while (process_inter == 0);
37         for (i = 0; i < 5; i++) {
38             usleep(100);
39             printf("this is child process i=%d\n", i);
40         }
41     } else if (pid > 0) {    //
42         int i = 0;
43         for (i = 0; i < 5; i++) {
44             usleep(100);
45             printf("this is parent process i=%d\n", i);
46         }
47         process_inter = 1;
48         sleep(2);
49         write(fd[1], &process_inter, sizeof(process_inter));
50     }
51
52     while(1);
53     return 0;
54 }

```

## JZ2440

- 

```
arm-linux-gcc comm_fork.c -o comm_fork
```

- NFS



```
cp comm_fork /work/nfs_root/first_fs
```

- 

```
./comm_fork
```

38 2s

2s

ProcessCommunication\_Image018

Image not found or type unknown

# 4.3.2

## 4.3.2.1

Linux 7

	'-' open
	'd' mkdir
	'l', la -s
( )	'p' mkfifo
socket	's' socket
	'c'
	'b'

## 4.3.2.2

```
int mkfifo(const char * filename, mode_t mode)
{
    umask(0);
    return 0;
}
```

mkfifo

## 4.3.2.3

( jz2440\process\_pipe\4th\_create\_myfifo\create\_myfifo.c)

```

01 /*****
02 *      1.
03 *
04 *
05 *
06 *
07 * -----
08 * 2020/05/16      V1.0      zh(ryan)
09 *****/
10
11 #include <stdio.h>
12 #include <stdlib.h>
13 #include <unistd.h>
14 #include <sys/types.h>
15
16 int main(int argc, char *argv[])
17 {
18     int ret;
19
20     ret = mkfifo("./myfifo", 0777);    //      777
21     if (ret < 0) {
22         printf("create myfifo fail\n");
23         return -1;
24     }
25     printf("create myfifo sucess\n");
26
27     return 0;
28 }

```

## JZ2440

- 

```
arm-linux-gcc create_myfifo.c -o create_myfifo
```

- NFS

```
cp create_myfifo /work/nfs_root/first_fs
```

•

```
./create_myfifo
```

myfifo      "-p"

ProcessCommunication\_Image019

Image not found or type unknown

**2**

1 ( jz2440\process\_pipe\5th\_myfifo\_comm\5nd\_named\_pipe.c)

```
01 /*****
02 *      1. 1      3rd_fifo  0777
03      2.
04 *
05 *
06 *
07 *
08 * -----
09 * 2020/05/16      V1.0      zh(ryan)
10 *****/
11
12 #include <stdio.h>
13 #include <stdlib.h>
14 #include <unistd.h>
15 #include <sys/types.h>
16 #include <fcntl.h>
17
18 int main(int argc, char *argv[])
19 {
20     int i, ret, fd;
21     char p_flag = 0;
22
23     /*      */
24     if (access("./3rd_fifo", 0) < 0) { //      ,
25         ret = mkfifo("./3rd_fifo", 0777);
26         if (ret < 0) {
27             printf("create named pipe fail\n");
28             return -1;
```

```

29     }
30     printf("create named pipe sucess\n");
31 }
32
33 /*          */
34 fd=open("./3rd_fifo", O_WRONLY);
35 if (fd < 0) {
36     printf("open 3rd_fifo fail\n");
37     return -1;
38 }
39 printf("open 3rd_fifo sucess\n");
40
41 for (i = 0; i < 5; i++) {
42     printf("this is first process i=%d\n", i);
43     usleep(100);
44 }
45 p_flag = 1;
46 sleep(5);
47 write(fd, &p_flag, sizeof(p_flag));
48
49 while(1);
50 return 0;
51 }

```

## 2 ( jz2440\process\_pipe\5th\_myfifo\_comm\5nd\_named\_pipe\_2.c)

```

01 /*****
02 *      1.
03      2.
04 *
05 *
06 *
07 *
08 * -----
09 * 2020/05/16      V1.0      zh(ryan)
10 *****/
11
12 #include <stdio.h>
13 #include <stdlib.h>
14 #include <unistd.h>
15 #include <sys/types.h>

```

```

16 #include <fcntl.h>
17
18 int main(int argc, char *argv[])
19 {
20     int i;
21     int fd=open("./3rd_fifo", O_RDONLY);
22     char p_flag = 0;
23
24     if (fd < 0) {
25         printf("open 3rd_fifo fail\n");
26         return -1;
27     }
28
29     printf("open 3rd_fifo sucess\n");
30     read(fd, &p_flag, sizeof(p_flag));
31     while(!p_flag);
32     for (i = 0; i < 5; i++) {
33         printf("this is second process i=%d\n", i);
34         usleep(100);
35     }
36
37     while(1);
38     return 0;
39 }

```

## JZ2440

- 

```
arm-linux-gcc 5nd_named_pipe.c -o 5nd_named_pipe
```

```
arm-linux-gcc 5nd_named_pipe_2.c -o 5nd_named_pipe_2
```

- NFS

```
cp 5nd_named_pipe /work/nfs_root/first_fs
```

```
cp 5nd_named_pipe_2 /work/nfs_root/first_fs
```

-

```
./5nd_named_pipe &

./5nd_named_pipe_2 &
```

ProcessCommunication\_Image020

Image not found or type unknown

# 4.4 IPC

IPC

IPC

IPC

- key key IPC\_PRIVATE key ftok
- key IPC IPC IPC ID IPC\_id shm\_id msg\_id :
- IPC\_id IPC IPC shmctrl shmat shmdt msgctrl msgsnd msgrcv

ProcessCommunication\_Image021

Image not found or type unknown

key IPC\_id shm\_id/msg\_id/sem\_id

IPC

IPC ID IPC\_id IPC\_id ID IPC IPC

ProcessCommunication\_Image022

Image not found or type unknown

ftok

```
: char ftok(const char *path, char key)

path

key

key -1
```

ftok key IPC

IPC\_PRIVATE

ftok

key

key

key

key

ftok

key

## 4.4.1

### 4.4.1.1

Linux

ProcessCommunication\_Image023

Image not found or type unknown

- 
- key ID
- 

### 4.4.1.2

```
    : int shmget(key_t key, int size, int shmflg)

    : #include <sys/shm.h>

    : key: IPC_PRIVATE    ftok

    □ IPC_PRIVATE    key    , 0

    □ size :

    □ shmflg :  open

    ID -1
```

**1 jz2440\process\_ipc\1st\_shm\1st\_shm.c**

```

01 /*****
02 *      1.  IPC_PRIVATE
03 *
04 *
05 *
06 *
07 * -----
08 * 2020/05/16      V1.0      zh(ryan)
09 *****/
10
11 #include <stdio.h>
12 #include <stdlib.h>
13 #include <unistd.h>
14 #include <sys/types.h>
15 #include <sys/shm.h>
16 #include <signal.h>
17
18 int main(int argc, char *argv[])
19 {
20     int shmid;
21
22     shmid = shmget(IPC_PRIVATE, 128, 0777);
23     if (shmid < 0) {
24         printf("create shared memory fail\n");
25         return -1;
26     }
27     printf("create shared memory sucess, shmid = %d\n", shmid);
28     system("ipcs -m");
29     return 0;
30 }

```

## JZ2440

- 

```
arm-linux-gcc 1st_shm.c -o 1st_shm
```

- NFS

```
cp 1st_shm /work/nfs_root/first_fs
```

-



```
./1st_shm
```

## ProcessCommunication\_Image024

Image not found or type unknown

22440\process\_ipc\1st\_shm\2nd\_shm.c

fotk      key

```
01 /*****
02 *      1.  ftok      key
03 *
04 *
05 *
06 *
07 * -----
08 * 2020/05/16      V1.0      zh(ryan)
09 *****/
10
11 #include <stdio.h>
12 #include <stdlib.h>
13 #include <unistd.h>
14 #include <sys/types.h>
15 #include <sys/shm.h>
16 #include <signal.h>
17
18 int main(int argc, char *argv[])
19 {
20     int shmid;
21     int key;
22
23     key = ftok("./a.c", 'a'); //      key
24     if (key < 0) {
25         printf("create key fail\n");
26         return -1;
27     }
28     printf("create key sucess key = 0x%X\n",key);
29
30     shmid = shmget(key, 128, IPC_CREAT | 0777);
```

```

31     if (shmid < 0) {
32         printf("create shared memory fail\n");
33         return -1;
34     }
35     printf("create shared memory sucess, shmid = %d\n", shmid);
36     system("ipcs -m");
37     return 0;
38 }

```

## JZ2440

- 

```
arm-linux-gcc 2nd_shm.c -o 2nd_shm
```

- NFS

```
cp 2nd_shm /work/nfs_root/first_fs
```

- 

2nd\_shm                      a.c   jz2440

```
touch a.c
```

key      0x610d0169.

```
./2nd_shm
```

## ProcessCommunication\_Image025

Image not found or type unknown

## 4.4.1.3

### shmat

```
void *shmat(int shmid, const void *shmaddr, int shmflg)
```

shmid ID

```

[] shmaddr      NULL

      shmflg SHM_RDONLY

      0

      NULL

```

jz2440\process\_ipc\1st\_shm\3nd\_shm.c

```

01 /*****
02 *      1.      shmat
03      2.
04      3.
05 *
06 *
07 *
08 *
09 * -----
10 * 2020/05/16      V1.0      zh(ryan)
11 *****/
12 #include <stdio.h>
13 #include <stdlib.h>
14 #include <unistd.h>
15 #include <sys/types.h>
16 #include <sys/shm.h>
17 #include <signal.h>
18
19 int main(int argc, char *argv[])
20 {
21     int shmid;
22     int key;
23     char *p;
24
25     key = ftok("./a.c", 'b');
26     if (key < 0) {
27         printf("create key fail\n");
28         return -1;
29     }
30     printf("create key sucess key = 0x%X\n",key);

```

```

31
32     shmidx = shmget(key, 128, IPC_CREAT | 0777);
33     if (shmidx < 0) {
34         printf("create shared memory fail\n");
35         return -1;
36     }
37     printf("create shared memory success, shmidx = %d\n", shmidx);
38     system("ipcs -m");
39
40     p = (char *)shmat(shmidx, NULL, 0);
41     if (p == NULL) {
42         printf("shmat fail\n");
43         return -1;
44     }
45     printf("shmat success\n");
46
47     // console
48     fgets(p, 128, stdin);
49
50     //
51     printf("share memory data: %s\n", p);
52
53     //
54     printf("share memory data: %s\n", p);
55     return 0;
56 }

```

## JZ2440

- 

```
arm-linux-gcc 3nd_shm.c -o 3nd_shm
```

- NFS

```
cp 3nd_shm /work/nfs_root/first_fs
```

- 

3nd\_shm                      a.c jz2440

```
touch a.c
```

```
./3rd_shm
```

## ProcessCommunication\_Image026

Image not found or type unknown

console "hello linux"

## ProcessCommunication\_Image027

Image not found or type unknown

51 54

## shmdt

```
int shmdt(const void *shmaddr)
```

```
shmat
```

```
: 0 -1
```

jz2440\process\_ipc\1st\_shm\4th\_shm.c

```
01 /*****
02 *      1.      shmat
03      2.
04      3.
05      4. shmdt
06 *
07 *
08 *
09 *
10 * -----
11 * 2020/05/16      V1.0      zh(ryan)
12 *****/
13 #include <stdio.h>
14 #include <stdlib.h>
15 #include <unistd.h>
16 #include <sys/types.h>
17 #include <sys/shm.h>
```

```
18 #include <signal.h>
19 #include <string.h>
20
21 int main(int argc, char *argv[])
22 {
23     int shmid;
24     int key;
25     char *p;
26
27     key = ftok("./a.c", 'b');
28     if (key < 0) {
29         printf("create key fail\n");
30         return -1;
31     }
32     printf("create key sucess key = 0x%X\n", key);
33
34     shmid = shmget(key, 128, IPC_CREAT | 0777);
35     if (shmid < 0) {
36         printf("create shared memory fail\n");
37         return -1;
38     }
39     printf("create shared memory sucess, shmid = %d\n", shmid);
40     system("ipcs -m");
41
42     p = (char *)shmat(shmid, NULL, 0);
43     if (p == NULL) {
44         printf("shmat fail\n");
45         return -1;
46     }
47     printf("shmat sucess\n");
48
49     //write share memory
50     fgets(p, 128, stdin);
51
52     //start read share memory
53     printf("share memory data: %s\n", p);
54
55     //start read share memory again
56     printf("share memory data: %s\n", p);
57
58     //
```

```

59     shmctl(p);
60
61     memcpy(p, "abcd", 4); //      segment fault
62     return 0;
63 }

```

## JZ2440

- 

```
arm-linux-gcc 4th_shm.c -o 4th_shm
```

- NFS

```
cp 4th_shm /work/nfs_root/first_fs
```

- 

4th\_shm.c                      a.c   jz2440

```
touch a.c
```

,                      61       Segmentation fault

```
./4th_shm
```

## ProcessCommunication\_Image028

Image not found or type unknown

## shmctl

```

int shmctl(int shmid, int cmd, struct shmid_ds *buf)

: shmid :

[] cmd : IPC_START (      ) ---      ipcs -m

      IPC_SET(      )

[] IPC_RMID (      ) ---      ipcrm -m

[] buf :   IPC_START/IPC_SET   /

```

: 0 -1

jz2440\process\_ipc\1st\_shm\5th\_shm.c

```
01 /*****
02 *      1.          shmat
03          2.
04          3.
05          4.  shmdt
06          5.  shmctl
07 *
08 *
09 *
10 *
11 * -----
12 * 2020/05/16      V1.0      zh(ryan)
13 *****/
14
15 #include <stdio.h>
16 #include <stdlib.h>
17 #include <unistd.h>
18 #include <sys/types.h>
19 #include <sys/shm.h>
20 #include <signal.h>
21 #include <string.h>
22
23 int main(int argc, char *argv[])
24 {
25     int shmid;
26     int key;
27     char *p;
28
29     key = ftok("./a.c", 'b');
30     if (key < 0) {
31         printf("create key fail\n");
32         return -1;
33     }
34     printf("create key sucess key = 0x%X\n",key);
35
36     shmid = shmget(key, 128, IPC_CREAT | 0777);
```



```

37     if (shmid < 0) {
38         printf("create shared memory fail\n");
39         return -1;
40     }
41     printf("create shared memory sucess, shmid = %d\n", shmid);
42     system("ipcs -m");
43
44     p = (char *)shmat(shmid, NULL, 0);
45     if (p == NULL) {
46         printf("shmat fail\n");
47         return -1;
48     }
49     printf("shmat sucess\n");
50
51     //write share memory
52     fgets(p, 128, stdin);
53
54     //start read share memory
55     printf("share memory data: %s\n", p);
56
57     //start read share memory again
58     printf("share memory data: %s\n", p);
59
60     //
61     shmdt(p);
62
63     //memcpy(p, "abcd", 4); //      segment fault
64
65     shmctl(shmid, IPC_RMID, NULL);
66     system("ipcs -m");
67     return 0;
68 }

```

## JZ2440

- 

```
arm-linux-gcc 5th_shm.c -o 5th_shm
```

- NFS

```
cp 5th_shm /work/nfs_root/first_fs
```

- 

```
touch a.c
```

42

66

```
./4th_shm
```

ProcessCommunication\_Image029

Image not found or type unknown

## 4.4.1.4

1. /

2.

3.

4.

5.

1. ipcs -l cat /proc/sys/kernel/shmmax

2. shmctl ( shmdt )

jz2440\process\_ipc\1st\_shm\6th\_shm.c

```
01 /*****
```

```
02 * 1. key IPC_PRIVATE
```

```
03 2.
```

```
04 3.
```

```
05 4.
```

```
06 5.
```

```
07          6.
08  *
09  *
10  *
11  *
12  * -----
13  * 2020/05/16      V1.0      zh(ryan)
14  *****/
15
16 #include <stdio.h>
17 #include <stdlib.h>
18 #include <unistd.h>
19 #include <sys/types.h>
20 #include <sys/shm.h>
21 #include <signal.h>
22 #include <string.h>
23
24 void myfun(int signum)
25 {
26     return;
27 }
28
29 int main(int argc, char *argv[])
30 {
31     int shmid;
32     int key;
33     char *p;
34     int pid;
35
36
37     shmid = shmget(IPC_PRIVATE, 128, IPC_CREAT | 0777);
38     if (shmid < 0) {
39         printf("create shared memory fail\n");
40         return -1;
41     }
42     printf("create shared memory sucess, shmid = %d\n", shmid);
43
44     pid = fork();
45     if (pid > 0) {    //
46         signal(SIGUSR2, myfun);
47         p = (char *)shmat(shmid, NULL, 0);
```

```

48     if (p == NULL) {
49         printf("shmat fail\n");
50         return -1;
51     }
52     printf("parent process shmat sucess\n");
53     while (1) {
54         //
55         printf("parent process begin to write memory data:");
56         fgets(p, 128, stdin);
57         kill(pid, SIGUSR1);    //
58         pause();              //
59     }
60 }
61 if (pid == 0) { //
62     signal(SIGUSR1, myfun);
63     p = (char *)shmat(shmid, NULL, 0);
64     if (p == NULL) {
65         printf("shmat fail\n");
66         return -1;
67     }
68     printf("child process shmat sucess\n");
69     while (1) {
70         pause(); //
71         //
72         printf("child process read share memory data: %s\n", p);
73         kill(getppid(), SIGUSR2);
74     }
75 }
76
77 //
78 shmdt(p);
79
80 //memcpy(p, "abcd", 4); //      segment fault
81
82 shmctl(shmid, IPC_RMID, NULL);
83 system("ipcs -m");
84 return 0;
85 }

```

- 

```
arm-linux-gcc 6th_shm.c -o 6th_shm
```

- NFS

```
cp 6th_shm /work/nfs_root/first_fs
```

- 

```
./6th_shm
```

## ProcessCommunication\_Image030

Image not found or type unknown

server jz2440\process\_ipc\1st\_shm\7th\_shm\_1.c

```
01 /*****
02 *      1. server   ftok   key   key
03      2.
04      3. server      client
05 *
06 *
07 *
08 *
09 * -----
10 * 2020/05/16      V1.0      zh(ryan)
11 *****/
12
13 #include <stdio.h>
14 #include <stdlib.h>
15 #include <unistd.h>
16 #include <sys/types.h>
17 #include <sys/shm.h>
18 #include <signal.h>
19 #include <string.h>
20
21 struct mybuf
22 {
```

```
23     int pid;
24     char buf[124];
25 };
26
27 void myfun(int signum)
28 {
29     return;
30 }
31
32 int main(int argc, char *argv[])
33 {
34     int shmid;
35     int key;
36     struct mybuf *p;
37     int pid;
38
39     key = ftok("./a.c", 'a');
40     if (key < 0) {
41         printf("create key fail\n");
42         return -1;
43     }
44     printf("create key sucess\n");
45
46     shmid = shmget(key, 128, IPC_CREAT | 0777);
47     if (shmid < 0) {
48         printf("create shared memory fail\n");
49         return -1;
50     }
51     printf("create shared memory sucess, shmid = %d\n", shmid);
52
53     signal(SIGUSR2, myfun);
54     p = (struct mybuf *)shmat(shmid, NULL, 0);
55     if (p == NULL) {
56         printf("shmat fail\n");
57         return -1;
58     }
59     printf("parent process shmat sucess\n");
60
61     p->pid = getpid(); // server  pid
62     pause();           // client  server pid
63     pid=p->pid;         // client
```

```

64
65     while (1) {
66         //write share memory
67         printf("parent process begin to write memory data\n");
68         fgets(p->buf, 124, stdin);
69         kill(pid, SIGUSR1);    // client    client
70         pause();               // client
71     }
72
73     //
74     shmdt(p);
75
76     shmctl(shmid, IPC_RMID, NULL);
77     system("ipcs -m");
78     return 0;
79 }

```

client          jz2440\process\_ipc\1st\_shm\7th\_shm\_2.c

```

01 /*****
02 *      1.client   ftok   key   key
03          2.client   server
04          3.
05 *
06 *
07 *
08 *
09 * -----
10 * 2020/05/16      V1.0      zh(ryan)
11 *****/
12
13 #include <stdio.h>
14 #include <stdlib.h>
15 #include <unistd.h>
16 #include <sys/types.h>
17 #include <sys/shm.h>
18 #include <signal.h>
19 #include <string.h>
20
21 struct mybuf
22 {

```

```
23     int pid;
24     char buf[124];
25 };
26
27 void myfun(int signum)
28 {
29     return;
30 }
31
32 int main(int argc, char *argv[])
33 {
34     int shmid;
35     int key;
36     struct mybuf *p;
37     int pid;
38
39     key = ftok("./a.c", 'a');
40     if (key < 0) {
41         printf("create key fail\n");
42         return -1;
43     }
44     printf("create key sucess\n");
45
46     shmid = shmget(key, 128, IPC_CREAT | 0777);
47     if (shmid < 0) {
48         printf("create shared memory fail\n");
49         return -1;
50     }
51     printf("create shared memory sucess, shmid = %d\n", shmid);
52
53     signal(SIGUSR1, myfun);
54     p = (struct mybuf *)shmat(shmid, NULL, 0);
55     if (p == NULL) {
56         printf("shmat fail\n");
57         return -1;
58     }
59     printf("client process shmat sucess\n");
60
61     // get server pid
62     //read share memory
63     pid = p->pid;
```



```

64    // write client pid to share memory
65    p->pid = getpid();
66    kill(pid, SIGUSR2);    // tell server process to read data
67
68    //client start to read share memory
69
70    while (1) {
71        pause();            // wait server process write share memory
72        printf("client process read data: %s\n", p->buf); // read data
73        kill(pid, SIGUSR2); // server can write share memory
74    }
75
76    //
77    shmdt(p);
78
79    shmctl(shmid, IPC_RMID, NULL);
80    system("ipcs -m");
81    return 0;
82 }

```

console

server

client

console

telnet console

## 4.4.2

### 4.4.2.1

cat/proc/sys/kernel/msgmax

msgqid\_ds

msqid\_ds

msqid\_ds.msc

ProcessCommunication\_Image031

Image not found or type unknown

### 4.4.2.2

- 1.
- 2.
- 3.

### 4.4.2.3

**msgget**

```

#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
int msgget(key_t key, int flag)
key          key
flag
ID          -1

```

## msgctl

```

int msgctl(int msgqid, int cmd, struct msqid_ds *buf)
msgqid      ID
cmd IPC_STAT      buf
      IPC_SET      buf
      IPC_RMID
      buf
0          -1

```

## msgsnd

```

#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
int msgsnd(int msgqid, const void *msgp, size_t size, int flag)
msgqid      ID
msgp          msgbuf
struct msgbuf {
    long mtype;    //
    char mtext[N]; //
};
size
flag IPC_NOWAIT
0
0          -1

```

## msgrcv

```

int msgrcv(int msgqid, void *msgp, size_t size, long msgtype, int flag)
msgqid      ID
    msgp
    size
    msgtype  0
            0          msgtype
            0          msgtype
    flag IPC_NOWAIT
        0
        -1

```

## 4.4.2.4

server jz2440\process\_ipc\2nd\_shm\write\_msg.c

```

01 /*****
02 *      1. server
03 *
04 *
05 *
06 *
07 * -----
08 * 2020/05/16      V1.0      zh(ryan)
09 *****/
10
11 #include <stdio.h>
12 #include <stdlib.h>
13 #include <unistd.h>
14 #include <sys/types.h>
15 #include <sys/msg.h>
16 #include <signal.h>
17 #include <string.h>
18
19 struct msgbuf {
20     long type;          //
21     char voltage[124];  //
22     char ID[4];
23 };
24

```

```

25 int main(int argc, char *argv[])
26 {
27     int msgid, readret, key;
28     struct msgbuf sendbuf;
29
30     key = ftok("./a.c", 'a');
31     if (key < 0){
32         printf("create key fail\n");
33         return -1;
34     }
35     msgid = msgget(key, IPC_CREAT|0777);
36     if (msgid < 0) {
37         printf("create msg queue fail\n");
38         return -1;
39     }
40     printf("create msg queue sucess, msgid = %d\n", msgid);
41     system("ipcs -q");
42
43     // write message queue
44     sendbuf.type = 100;
45     while(1) {
46         memset(sendbuf.voltage, 0, 124); //clear send buffer
47         printf("please input message:");
48         fgets(sendbuf.voltage, 124, stdin);
49         //start write msg to msg queue
50         msgsnd(msgid, (void *)&sendbuf, strlen(sendbuf.voltage), 0);
51     }
52
53     return 0;
54 }

```

client jz2440\process\_ipc\2nd\_shm\read\_msg.c

```

01 /*****
02 *      1.client
03 *
04 *
05 *
06 *
07 * -----
08 * 2020/05/16      V1.0      zh(ryan)

```

```

09  *****/
10
11 #include <stdio.h>
12 #include <stdlib.h>
13 #include <unistd.h>
14 #include <sys/types.h>
15 #include <sys/msg.h>
16 #include <signal.h>
17 #include <string.h>
18
19 struct msgbuf {
20     long type;          //
21     char voltage[124];  //
22     char ID[4];
23 };
24
25 int main(int argc, char *argv[])
26 {
27     int msgid, key;
28     struct msgbuf readbuf;
29
30     key = ftok("./a.c", 'a');
31     if (key < 0){
32         printf("create key fail\n");
33         return -1;
34     }
35     msgid = msgget(key, IPC_CREAT|0777);
36     if (msgid < 0) {
37         printf("create msg queue fail\n");
38         return -1;
39     }
40     printf("create msg queue sucess, msgid = %d\n", msgid);
41     system("ipcs -q");
42
43     // read message queue
44     while(1) {
45         memset(readbuf.voltage, 0, 124); //clear recv buffer
46         //start read msg to msg queue
47         msgrcv(msgid, (void *)&readbuf, 124, 100, 0);
48         printf("recv data from message queue: %s", readbuf.voltage);
49     }

```

```
50
51     return 0;
52 }
```

## JZ2440

- 

```
arm-linux-gcc write_msg.c -o write_msg
arm-linux-gcc read_msg.c -o read_msg
```

- NFS

```
cp write_msg /work/nfs_root/first_fs
cp read_msg /work/nfs_root/first_fs
```

- 

read\_msg      write\_msg      console      client

```
./read_msg &
./ write_msg
```

## ProcessCommunication\_Image032

Image not found or type unknown

# 4.4.3

## 4.4.3.1      P   V

P                  V

## 4.4.3.2

P/V                  /                  semid\_ds

## ProcessCommunication\_Image033

Image not found or type unknown

POSIX                  POSIX                  IPC

	(POSIX)	(IPC )
	sem_t sem1	semget
	sem_init	semctl
P	sem_wait	semop
V	sem_post	semop

IPC

POSIX

1.

A

B

1

2

A

1

2
2.

B

2

1
3.

A

1

2

B

2

1
4.

B

2

A

1

B

1

2

A

2

1

### 4.4.3.3

#### semget

```
#includde <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
: int semget(key_t key, int nsems, int semflag)
key          key
nsems
semflag
ID    -1
```

#### semctl

```
#includde <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
: int semctl(int semid, int semnum, int cmd, ...union semun arg)

semid      id
semnum
cmd GETVAL
SETVAL
IPC_RMID
```

```

union semun arg: union semun {
    int          val;      /* Value for SETVAL */
    struct semid_ds *buf;   /* Buffer for IPC_STAT, IPC_SET */
    unsigned short *array; /* Array for GETALL, SETALL */
    struct seminfo *__buf;  /* Buffer for IPC_INFO (Linux-specific) */
};

ID -1

```

## semop

```

p/v
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>

: int semop(int semid, struct sembuf *opsptr, size_t nops)
semid id
opsptr struct sembuf{
    short sem_num;    //
    short sem_op;     //0:      0 1:      V -1:      P
    short sem_flg;    //0: IPC_NOWAIT, SEM_UNDO
}

nops
ID -1

```

## 4.4.3.4 /

jz2440\process\_ipc\3rd\_shm\share\_sysv.c

```

01 /*****
02 *      1.      .
03      2.      .
04      3.      quit      .
05 *
06 *
07 *
08 *
09 * -----
10 * 2020/05/16      V1.0      zh(ryan)
11 *****/
12
13 #include <stdio.h>

```



```

14 #include <stdlib.h>
15 #include <string.h>
16 #include <sys/ipc.h>
17 #include <sys/sem.h>
18 #include <sys/types.h>
19 #include <sys/shm.h>
20 #include <signal.h>
21 #include <unistd.h>
22
23 #define N 64
24 #define READ 0
25 #define WRITE 1
26
27 union semun {
28     int val;
29     struct semid_ds *buf;
30     unsigned short *array;
31     struct seminfo *__buf;
32 };
33
34 void init_sem(int semid, int s[], int n)
35 {
36     int i;
37     union semun myun;
38
39     for (i = 0; i < n; i++){
40         myun.val = s[i];
41         semctl(semid, i, SETVAL, myun);
42     }
43 }
44
45 void pv(int semid, int num, int op)
46 {
47     struct sembuf buf;
48
49     buf.sem_num = num;
50     buf.sem_op = op;
51     buf.sem_flg = 0;
52     semop(semid, &buf, 1);
53 }
54

```

```

55 int main(int argc, char *argv[])
56 {
57     int shmid, semid, s[] = {0, 1};
58     pid_t pid;
59     key_t key;
60     char *shmaddr;
61
62     key = ftok(".", 's');
63     if (key == -1){
64         perror("ftok");
65         exit(-1);
66     }
67
68     shmid = shmget(key, N, IPC_CREAT|0666);
69     if (shmid < 0) {
70         perror("shmget");
71         exit(-1);
72     }
73
74     semid = semget(key, 2, IPC_CREAT|0666);
75     if (semid < 0) {
76         perror("semget");
77         goto __ERROR1;
78     }
79     init_sem(semid, s, 2);
80
81     shmaddr = shmat(shmid, NULL, 0);
82     if (shmaddr == NULL) {
83         perror("shmat");
84         goto __ERROR2;
85     }
86
87     pid = fork();
88     if(pid < 0) {
89         perror("fork");
90         goto __ERROR2;
91     } else if (pid == 0) {
92         char *p, *q;
93         while(1) {
94             pv(semid, READ, -1);
95             p = q = shmaddr;

```

```

96 while (*q) {
97     if (*q != ' ') {
98         *p++ = *q;
99     }
100     q++;
101 }
102 *p = '\0';
103 printf("%s", shmaddr);
104 pv(semid, WRITE, 1);
105 }
106 } else {
107     while (1) {
108         pv(semid, WRITE, -1);
109         printf("input > ");
110         fgets(shmaddr, N, stdin);
111         if (strcmp(shmaddr, "quit\n") == 0) break;
112         pv(semid, READ, 1);
113     }
114     kill(pid, SIGUSR1);
115 }
116
117 __ERROR2:
118 semctl(semid, 0, IPC_RMID);
119 __ERROR1:
120 shmctl(shmid, IPC_RMID, NULL);
121 return 0;
122 }

```

## JZ2440

- 

```
arm-linux-gcc share_sysv.c -o share_sysv
```

- NFS

```
cp share_sysv /work/nfs_root/first_fs
```

- 

console

```
./share_sysv
```

ProcessCommunication\_Image034

Image not found or type unknown

## 4.5

### 4.5.1

1. GPIO CPU
- 2.
3. Linux
4. Linux unix

### 4.5.2

ProcessCommunication\_Image035

Image not found or type unknown

### 4.5.3

**kill**

```
#include <unistd.h>

#include <signal.h>

int kill(pid_t pid, int sig);

pid

0 -1 INIT

sig

0 EOF
```

## raise

```
#include <unistd.h>

#include <signal.h>

int raise(int sig);

sig

0      EOF
```

## alarm

```
#include <unistd.h>

#include <signal.h>

int alarm(unsigned int seconds);

seconds

EOF
```

## pause

```
sleep

#include <unistd.h>

#include <signal.h>

int pause(void);

0      EOF
```

## signal

```
kill killall

#include <unistd.h>
```

```

#include <signal.h>

void (*signal(int signo, void(*handler)(int)))(int)

signo

handler

0      EOF

```

## 4.5.4

jz2440\process\_single\send\_single.c

```

01 /*****
02  *      1.
03  *
04  *
05  *
06  *
07  * -----
08  * 2020/05/16      V1.0      zh(ryan)
09  *****/
10
11 #include <stdio.h>
12 #include <stdlib.h>
13 #include <unistd.h>
14 #include <sys/types.h>
15 #include <signal.h>
16
17 void handler(int signo)
18 {
19     switch (signo) {
20     case SIGINT:
21         printf("I have got SIGINT\n");
22         break;
23     case SIGQUIT:
24         printf("I have got SIGQUIT\n");
25

```

```

26     break;
27 }
28 default:
29     printf("don't respond to this signal[%d]\n", signo);
30     exit(0);
31 }
32 }
33
34 int main(int argc, char *argv[])
35 {
36     signal(SIGINT, handler);
37     signal(SIGQUIT, handler);
38     while (1)
39         pause();
40     return 0;
41 }

```

## JZ2440

- 

```
arm-linux-gcc send_single.c -o send_single
```

- NFS

```
cp send_single /work/nfs_root/first_fs
```

- 

```
./send_single
```

## 4.6 socket

### 4.6.1 socket

QQ QQ QQ socket

Socket UNIX Unix/Linux “ ” “ open -> write/read -> close”

client

socket

**listen**



socket

server

```
#include <sys/types.h>
```

```
#include <sys/socket.h>
```

```
: int listen(int sockfd, int backlog);
```

```
sockfd      socket()
```

```
backlog server      client
```

```
ID      -1
```

## accept

client server

```
#include <sys/types.h>
```

```
#include <sys/socket.h>
```

```
: int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);
```

```
sockfd      socket()
```

```
addr      client
```

```
addrlen:   client
```

```
ID      -1
```

## connect

client server client

```
#include <sys/types.h>
```

```
#include <sys/socket.h>
```

```
: int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
```

```
sockfd      socket()

addr        server

addr len:   server

ID          - 1
```

## send

```
#include <sys/types.h>

#include <sys/socket.h>

: ssize_t send(int sockfd, const void *buf, size_t len, int flags);

sockfd      socket

buf:

len:

flags:      0

MSG_DONTRoute

MSG_DONTWAIT

MSG_OOB

MSG_PEEK

MSG_WAITALL

ID          - 1
```

## recv

```
#include <sys/types.h>
```

```
#include <sys/socket.h>
```

```
: ssize_t recv(int sockfd, void *buf, size_t len, int flags);
```

```
sockfd
```

```
□ buf:
```

```
□ len:
```

```
□ flags: 0
```

```
MSG_DONTRROUTE
```

```
MSG_DONTWAIT
```

```
MSG_OOB
```

```
MSG_PEEK
```

```
MSG_WAITALL
```

```
ID - 1
```

## 4.6.3 socket

Server

1. socket

2. socket

3.

4.

5. /

## Client

1. socket

2. socket

3.

4. /

server jz2440\process\_socket\server.c

```
01 /*****
02 *      1.server  client          client
03 *
04 *
05 *
06 *
07 * -----
08 * 2020/05/16      V1.0      zh(ryan)
09 *****/
10
11 #include <stdio.h>
12 #include <stdlib.h>
13 #include <unistd.h>
14 #include <sys/types.h>
15 #include <sys/stat.h>
16 #include <string.h>
17 #include <arpa/inet.h>
18 #include <sys/un.h>
19
20 int main(int argc, char *argv[])
21 {
22     int lfd ,ret, cfd;
23     struct sockaddr_un serv, client;
24     socklen_t len = sizeof(client);
25     char buf[1024] = {0};
```

```
26     int recvlen;
27
28     // socket
29     lfd = socket(AF_LOCAL, SOCK_STREAM, 0);
30     if (lfd == -1) {
31         perror("socket error");
32         return -1;
33     }
34
35     //
36     unlink("server.sock");
37
38     // server
39     serv.sun_family = AF_LOCAL;
40     strcpy(serv.sun_path, "server.sock");
41
42     //
43     ret = bind(lfd, (struct sockaddr *)&serv, sizeof(serv));
44     if (ret == -1) {
45         perror("bind error");
46         return -1;
47     }
48
49     //
50     ret = listen(lfd, 36);
51     if (ret == -1) {
52         perror("listen error");
53         return -1;
54     }
55
56     //
57     cfd = accept(lfd, (struct sockaddr *)&client, &lcn);
58     if (cfd == -1) {
59         perror("accept error");
60         return -1;
61     }
62     printf("====client bind file: %s\n", client.sun_path);
63
64     while (1) {
65         recvlen = recv(cfd, buf, sizeof(buf), 0);
66         if (recvlen == -1) {
```

```

67         perror("recv error");
68         return -1;
69     } else if (recvlen == 0) {
70         printf("client disconnct...\n");
71         close(cfd);
72         break;
73     } else {
74         printf("server recv buf: %s\n", buf);
75         send(cfd, buf, recvlen, 0);
76     }
77 }
78
79 close(cfd);
80 close(lfd);
81 return 0;
82 }

```

client      jz2440\process\_socket\client.c

```

01 /*****
02 *      1.client                server
03 *
04 *
05 *
06 *
07 * -----
08 * 2020/05/16      V1.0      zh(ryan)
09 *****/
10
11 #include <stdio.h>
12 #include <stdlib.h>
13 #include <unistd.h>
14 #include <sys/types.h>
15 #include <sys/stat.h>
16 #include <string.h>
17 #include <arpa/inet.h>
18 #include <sys/un.h>
19
20 int main(int argc, char *argv[])
21 {
22     int lfd ,ret;

```

```
23     struct sockaddr_un serv, client;
24     socklen_t len = sizeof(client);
25     char buf[1024] = {0};
26     int recvlen;
27
28     // socket
29     lfd = socket(AF_LOCAL, SOCK_STREAM, 0);
30     if (lfd == -1) {
31         perror("socket error");
32         return -1;
33     }
34
35     //
36     unlink("client.sock");
37
38     //
39     client.sun_family = AF_LOCAL;
40     strcpy(client.sun_path, "client.sock");
41     ret = bind(lfd, (struct sockaddr *)&client, sizeof(client));
42     if (ret == -1) {
43         perror("bind error");
44         return -1;
45     }
46
47     // server
48     serv.sun_family = AF_LOCAL;
49     strcpy(serv.sun_path, "server.sock");
50     //
51     connect(lfd, (struct sockaddr *)&serv, sizeof(serv));
52
53     while (1) {
54         fgets(buf, sizeof(buf), stdin);
55         send(lfd, buf, strlen(buf)+1, 0);
56
57         recv(lfd, buf, sizeof(buf), 0);
58         printf("client recv buf: %s\n", buf);
59     }
60
61     close(lfd);
62     return 0;
63 }
```

## JZ2440

- 

```
arm-linux-gcc server.c -o server
```

```
arm-linux-gcc client.c -o client
```

- NFS

```
cp server /work/nfs_root/first_fs
```

```
cp client /work/nfs_root/first_fs
```

- 

server      client      client

```
./server &
```

```
./client
```

ProcessCommunication\_Image037

Image not found or type unknown

## 4.6.4 server client

server      client

1.              client              client              client
2.              client              client              client





# 5

## 5.1

### 5.1.1

-> -> -> -> ->

### 5.1.2

### 5.1.3 pthread\_t

PID " PID " tid pthread\_t

```
#include <pthread.h>
pthread_t pthread_self(void);
```

pthread\_self 1 tid

## 1 Phtread\_txex1.c

```
1#include <pthread.h>
2#include <stdio.h>
3
4int main()
5{
6    pthread_t tid = pthread_self(); //    tid
7    printf("tid = %lu\n", (unsigned long)tid);
8    return 0;
9}
```

POSIX pthread gcc gcc xxx.c -lpthread

## MultiThread\_Image001

Image not found or type unknown

# 5.1.4

```
#include <pthread.h>
int pthread_create(pthread_t *thread, const pthread_attr_t *attr, void *(*start_routine) (void
*), void *arg);
0
```

pthread\_create

pthread\_t

NULL

void \* v

## 2 Phtread\_txex2.c

```
1#include <pthread.h>
2#include <stdio.h>
3#include <unistd.h>
4#include <errno.h>
5
6void *fun(void *arg)
7{
8    printf("pthread_New = %lu\n", (unsigned long)pthread_self()); //    tid
```

```

9  }
10
11 int main()
12 {
13
14     pthread_t tid1;
15     int ret = pthread_create( &tid1, NULL, fun, NULL); //
16     if(ret != 0){
17         perror("pthread_create");
18         return -1;
19     }
20
21     /*tid_main    pthread_self    ID tid_new    pthread_create    tid    */
22     printf("tid_main = %lu tid_new = %lu \n", (unsigned long)pthread_self(), (unsigned long)tid1);
23
24     /*          sleep          */
25     sleep(1);
26
27     return 0;
28 }
29

```

## MultiThread\_Image002

Image not found or type unknown

```

pthread_create          pthread_create    tid          pthread_self
                                sleep
25

```

## MultiThread\_Image003

Image not found or type unknown

```

3    2

```

# 5.1.5

```

pthread_create()    void *          void *          3

```

### 3 Phtread\_txex3.c

```
1 #include <pthread.h>
2 #include <stdio.h>
3 #include <unistd.h>
4 #include <errno.h>
5
6 void *fun1(void *arg)
7 {
8     printf("%s: arg = %d Addr = %p\n", __FUNCTION__, *(int *)arg, arg);
9 }
10
11 void *fun2(void *arg)
12 {
13     printf("%s: arg = %d Addr = %p\n", __FUNCTION__, (int)(long)arg, arg);
14 }
15
16 int main()
17 {
18
19     pthread_t tid1, tid2;
20     int a = 50;
21     int ret = pthread_create(&tid1, NULL, fun1, (void *)&a); // a
22     if(ret != 0){
23         perror("pthread_create");
24         return -1;
25     }
27     ret = pthread_create(&tid2, NULL, fun2, (void *)(long)a); // a
28     if(ret != 0){
29         perror("pthread_create");
30         return -1;
31     }
32     sleep(1);
33     printf("%s: a = %d Add = %p \n", __FUNCTION__, a, &a);
34     return 0;
35 }
36
```

Image not found or type unknown

21

a

void

#### 4 Phtread\_txex4.c

```
1  #include <pthread.h>
2  #include <stdio.h>
3  #include <unistd.h>
4  #include <errno.h>
5
6  void *fun1(void *arg)
7  {
8      while(1){
9
10         printf("%s: arg = %d Addr = %p\n", __FUNCTION__, *(int *)arg, arg);
11         sleep(1);
12     }
13 }
14
15 void *fun2(void *arg)
16 {
17     while(1){
18
19         printf("%s: arg = %d Addr = %p\n", __FUNCTION__, (int)(long)arg, arg);
20         sleep(1);
21     }
22 }
23
24 int main()
25 {
26
27     pthread_t tid1, tid2;
28     int a = 50;
29     int ret = pthread_create(&tid1, NULL, fun1, (void *)&a);
30     if(ret != 0){
31         perror("pthread_create");
32     }
33     return -1;
```

```

33    }
34    sleep(1);
35    ret = pthread_create(&tid2, NULL, fun2, (void *) (long) a);
36    if (ret != 0) {
37        perror("pthread_create");
38        return -1;
39    }
40    while(1) {
41        a++;
42        sleep(1);
43        printf("%s: a = %d Add = %p \n", __FUNCTION__, a, &a);
44    }
45    return 0;
46 }
47

```

## MultiThread\_Image005

Image not found or type unknown

### 5 Phtread\_txex5.c

```

1  #include <pthread.h>
2  #include <stdio.h>
3  #include <unistd.h>
4  #include <string.h>
5  #include <errno.h>
6
7  struct Stu{
8      int Id;
9      char Name[32];
10     float Mark;
11 };
12
13 void *fun1(void *arg)
14 {
15     struct Stu *tmp = (struct Stu *) arg;
16     printf("%s: Id = %d Name = %s Mark = %.2f\n", __FUNCTION__, tmp->Id, tmp->Name, tmp->Mark);

```

```

17
18
19
20int main()
21{
22
23pthread_t tid1,tid2;
24struct Stu stu;
25stu.Id = 10000;
26strcpy( stu. Name," ZhangSan");
27stu. Mark = 94. 6;
28
29int ret = pthread_create( &tid1, NULL, fun1,( void *)&stu);
30if(ret != 0){
31perror("pthread_create");
32return -1;
33}
34printf("%s: Id = %d Name = %s Mark = %. 2f\n", __FUNCTION__, stu. Id, stu. Name, stu. Mark);
35sleep(1);
36return 0;
37
38

```

## MultiThread\_Image006

Image not found or type unknown

## 5.1.6

pthread\_exit

pthread\_cancel

```

#include <pthread.h>
void pthread_exit(void *retval);

```

void\*

NULL

```

#include <pthread.h>

```



```
int pthread_join(pthread_t thread, void **retval);
    0
```

tid

```
#define _GNU_SOURCE
#include <pthread.h>
int pthread_tryjoin_np(pthread_t thread, void **retval);
    0
```

0 pthread\_join

```
#include <pthread.h>
int pthread_cancel(pthread_t thread);
    0
```

tid tid 0

API API

## 6 Phtread\_txex6.c

```
1 #include <pthread.h>
2 #include <stdio.h>
3 #include <unistd.h>
4 #include <errno.h>
5
6 void *fun1(void *arg)
7 {
8     static int tmp = 0; // static pthread_join
9     //int tmp = 0;
10    tmp = *(int *)arg;
11    tmp+=100;
12    printf("%s: Addr = %p tmp = %d\n", __FUNCTION__, &tmp, tmp);
13    pthread_exit((void *)&tmp); // tmp void*
14 }
15
16
17 int main()
18 {
```

```

19
20    pthread_t tid1;
21    int a = 50;
22    void *Tmp = NULL; // pthread_join    void**
23    int ret = pthread_create( &tid1, NULL, fun1, ( void *) &a);
24    if( ret != 0){
25        perror("pthread_create");
26        return -1;
27    }
28    pthread_join( tid1, &Tmp);
29    printf(" %s: Addr = %p Val = %d\n", __FUNCTION__, Tmp, *(int *) Tmp);
30    return 0;
31 }
32

```

## MultiThread\_Image007

Image not found or type unknown

23

100

void\*

pthread\_join

slee

### 7 Phtread\_txex7.c

```

1  #define _GNU_SOURCE
2  #include <pthread.h>
3  #include <stdio.h>
4  #include <unistd.h>
5  #include <errno.h>
6
7  void *fun( void *arg)
8  {
9      printf("Pthread: %d Come !\n", (int )( long) arg+1);
10     pthread_exit( arg);
11 }
12
13
14 int main()
15 {
16     int ret, i, flag = 0;
17     void *Tmp = NULL;

```

```

18 pthread_t tid[3];
19 for(i = 0; i < 3; i++){
20     ret = pthread_create(&tid[i], NULL, fun, (void *) (long)i);
21     if(ret != 0){
22         perror("pthread_create");
23         return -1;
24     }
25 }
26 while(1){ // 3
27     for(i = 0; i < 3; i++){
28         if(pthread_tryjoin_np(tid[i], &Tmp) == 0){
29             printf("Pthread : %d exit ! \n", (int) (long) Tmp+1);
30             flag++;
31         }
32     }
33     if(flag >= 3) break;
34 }
35 return 0;
36 }
37

```

## MultiThread\_Image008

Image not found or type unknown

7

6

pthread\_tryjoin\_np

8 Phtread\_txex8.c

```

1 #define _GNU_SOURCE
2 #include <pthread.h>
3 #include <stdio.h>
4 #include <unistd.h>
5 #include <errno.h>
6
7 void *fun1(void *arg)
8 {
9     printf("Pthread: 1 come! \n");
10    while(1){

```

```

11    sleep(1);
12}
13
14
15void *fun2( void *arg)
16{
17    printf("Pthread: 2 come! \n");
18    pthread_cancel((pthread_t )( long) arg); //    1
19    pthread_exit(NULL);
20}
21
22int main()
23{
24    int ret,i, flag = 0;
25    void *Tmp = NULL;
26    pthread_t tid[ 2];
27    ret = pthread_create( &tid[ 0], NULL, fun1, NULL);
28    if( ret != 0){
29        perror("pthread_create");
30        return -1;
31    }
32    sleep(1);
33    ret = pthread_create( &tid[ 1], NULL, fun2, ( void *) tid[ 0]); //    1
34    if( ret != 0){
35        perror("pthread_create");
36        return -1;
37    }
38    while(1){ //                2
39        for(i = 0; i <2; i++){
40            if(pthread_tryjoin_np( tid[ i], NULL) == 0){
41                printf("Pthread :  %d exit ! \n", i+1);
42                flag++;
43            }
44        }
45        if( flag >= 2) break;
46    }
47    return 0;
48}
49

```

## MultiThread\_Image009

Image not found or type unknown

8 pthread\_cancel 27 33

# 5.2

## 5.2.1

“ ” 1 2 “ ”

9 Phtread\_txex9.c

```
1 #define _GNU_SOURCE
2 #include <pthread.h>
3 #include <stdio.h>
4 #include <unistd.h>
5 #include <errno.h>
6
7
8 int Num = 0;
9
10 void *fun1(void *arg)
11 {
12     while(Num < 3){
13         Num++;
14         printf("%s: Num = %d\n", __FUNCTION__, Num);
15         sleep(1);
16     }
17     pthread_exit(NULL);
18 }
19
20 void *fun2(void *arg)
21 {
22     while(Num > -3){
23         Num--;
24         printf("%s: Num = %d\n", __FUNCTION__, Num);
```

```

25    sleep(1);
26}
27pthread_exit(NULL);
28}
29
30int main()
31{
32    int ret;
33    pthread_t tid1, tid2;
34    ret = pthread_create(&tid1, NULL, fun1, NULL);
35    if(ret != 0){
36        perror("pthread_create");
37        return -1;
38    }
39    ret = pthread_create(&tid2, NULL, fun2, NULL);
40    if(ret != 0){
41        perror("pthread_create");
42        return -1;
43    }
44    pthread_join(tid1, NULL);
45    pthread_join(tid2, NULL);
46    return 0;
47}
48

```

## MultiThread\_Image010

Image not found or type unknown

pthread

## 5.2.2 API

```

#include <pthread.h>

int pthread_mutex_init(pthread_mutex_t *mutex,
const pthread_mutexattr_t *restrict attr);

0

```

```
pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
```

```

/
#include <pthread.h>
int pthread_mutex_lock(pthread_mutex_t *mutex);
int pthread_mutex_unlock(pthread_mutex_t *mutex);

```

```

lock    unlock                pthread_mutex_t    0    lock
        lock                unlock                pthread_cancel

```

```

#include <pthread.h>
int pthread_mutex_trylock(pthread_mutex_t *mutex);
    0

```

```

#include <pthread.h>
int pthread_mutex_destory(pthread_mutex_t *mutex);
    0

```

0

## 10 Phtread\_txex10.c

```

1  #define _GNU_SOURCE
2  #include <pthread.h>
3  #include <stdio.h>
4  #include <unistd.h>
5  #include <errno.h>
6
7  pthread_mutex_t mutex; //
8
9  int Num = 0; //
10
11 void *fun1(void *arg)
12 {
13     pthread_mutex_lock(&mutex); //
14     while(Num < 3){

```

```

15    Num++;
16    printf("%s: Num = %d\n", __FUNCTION__, Num);
17    sleep(1);
18}
19pthread_mutex_unlock(&mutex); //
20pthread_exit(NULL); //    pthread_join
21}
22
23void *fun2(void *arg)
24{
25    pthread_mutex_lock(&mutex); //
26    while(Num > -3){
27        Num--;
28        printf("%s: Num = %d\n", __FUNCTION__, Num);
29        sleep(1);
30    }
31    pthread_mutex_unlock(&mutex); //
32    pthread_exit(NULL); //    pthread_join
33}
34
35int main()
36{
37    int ret;
38    pthread_t tid1, tid2;
39    ret = pthread_mutex_init(&mutex, NULL); //
40    if(ret != 0){
41        perror("pthread_mutex_init");
42        return -1;
43    }
44    ret = pthread_create(&tid1, NULL, fun1, NULL); //    1
45    if(ret != 0){
46        perror("pthread_create");
47        return -1;
48    }
49    ret = pthread_create(&tid2, NULL, fun2, NULL); //    2
50    if(ret != 0){
51        perror("pthread_create");
52        return -1;
53    }
54    pthread_join(tid1, NULL); //    1
55    pthread_join(tid2, NULL); //    2

```



```

56    pthread_mutex_destroy( &mutex); //
57    return 0;
58}
59

```

## MultiThread\_Image011

Image not found or type unknown

while

## 5.2.3

sleep

11

11 Phtread\_txex11.c

```

1  #define _GNU_SOURCE
2  #include <pthread.h>
3  #include <stdio.h>
4  #include <unistd.h>
5  #include <errno.h>
6
7  void *fun1( void *arg)
8  {
9      printf( "%s: Pthread Come! \n", __FUNCTION__ );
10     pthread_exit( NULL );
11 }
12
13 void *fun2( void *arg)
14 {
15     printf( "%s: Pthread Come! \n", __FUNCTION__ );
16     pthread_exit( NULL );
17 }
18
19 void *fun3( void *arg)
20 {
21     printf( "%s: Pthread Come! \n", __FUNCTION__ );

```

```

22    pthread_exit(NULL);
23}
24
25int main()
26{
27    int ret;
28    pthread_t tid1, tid2, tid3;
29    ret = pthread_create(&tid1, NULL, fun1, NULL);
30    if(ret != 0){
31        perror("pthread_create");
32        return -1;
33    }
34    ret = pthread_create(&tid2, NULL, fun2, NULL);
35    if(ret != 0){
36        perror("pthread_create");
37        return -1;
38    }
39    ret = pthread_create(&tid3, NULL, fun3, NULL);
40    if(ret != 0){
41        perror("pthread_create");
42        return -1;
43    }
44    pthread_join(tid1, NULL);
45    pthread_join(tid2, NULL);
46    pthread_join(tid3, NULL);
47    return 0;
48}
49

```

## MultiThread\_Image012

Image not found or type unknown

## 5.2.4 API

```
#include <semaphore.h>
```

```
int sem_init(sem_t *sem, int pshared, unsigned int value);
    0
```

sem\_t                      0                      0                      1

```
PV
#include <pthread.h>
int sem_wait(sem_t *sem);
int sem_post(sem_t *sem);
    0
```

sem\_wait                                      “sem-1”                      “sem-1”

sem\_post                                      “sem+1”

2                      PV

```
#include <pthread.h>
int sem_trywait(sem_t *sem);
    0
```

sem\_wait

```
#include <pthread.h>
int sem_destory(sem_t *sem);
    0
```

## 12 Phtread\_txex12.c

```
1 #define _GNU_SOURCE
2 #include <pthread.h>
3 #include <stdio.h>
4 #include <unistd.h>
5 #include <errno.h>
6 #include <semaphore.h>
7
8 sem_t sem1, sem2, sem3; //
9
10 void *fun1(void *arg)
```

```

11{
12sem_wait(&sem1); // sem1 sem1-1
13printf("%s: Pthread Come! \n", __FUNCTION__);
14sem_post(&sem2); // sem2
15pthread_exit(NULL);
16}
17
18void *fun2( void *arg)
19{
20sem_wait(&sem2); // sem2 14 sem2-1
21printf("%s: Pthread Come! \n", __FUNCTION__);
22sem_post(&sem3); // sem3
23pthread_exit(NULL);
24}
25
26void *fun3( void *arg)
27{
28sem_wait(&sem3); // sem3 22 sem3-1
29printf("%s: Pthread Come! \n", __FUNCTION__);
30sem_post(&sem1); // sem1
31pthread_exit(NULL);
32}
33
34int main()
35{
36int ret;
37pthread_t tid1, tid2, tid3;
38ret = sem_init(&sem1, 0, 1); // 1
39if(ret < 0){
40perror("sem_init");
41return -1;
42}
43ret = sem_init(&sem2, 0, 0); // 2
44if(ret < 0){
45perror("sem_init");
46return -1;
47}
48ret = sem_init(&sem3, 0, 0); // 3
49if(ret < 0){
50perror("sem_init");
51return -1;

```

```

52{}
53ret = pthread_create( &tid1, NULL, fun1, NULL); // 1
54if(ret != 0){
55perror("pthread_create");
56return -1;
57}
58ret = pthread_create( &tid2, NULL, fun2, NULL); // 2
59if(ret != 0){
60perror("pthread_create");
61return -1;
62}
63ret = pthread_create( &tid3, NULL, fun3, NULL); // 3
64if(ret != 0){
65perror("pthread_create");
66return -1;
67}
68/* */
69pthread_join( tid1, NULL);
70pthread_join( tid2, NULL);
71pthread_join( tid3, NULL);
72
73/* */
74sem_destroy( &sem1);
75sem_destroy( &sem2);
76sem_destroy( &sem3);
77
78return 0;
79}
80

```

## MultiThread\_Image013

Image not found or type unknown

			1	sem_wait	sem_pos
38	1	43 48	2 3		sem_wait

# 5.3

MultiThread\_Image014

Image not found or type unknown

MultiThread\_Image015

Image not found or type unknown

PV

MultiThread\_Image016

Image not found or type unknown