

同济大学计算机系

操作系统课程实验报告



学 号 2154312

姓 名 郑博远

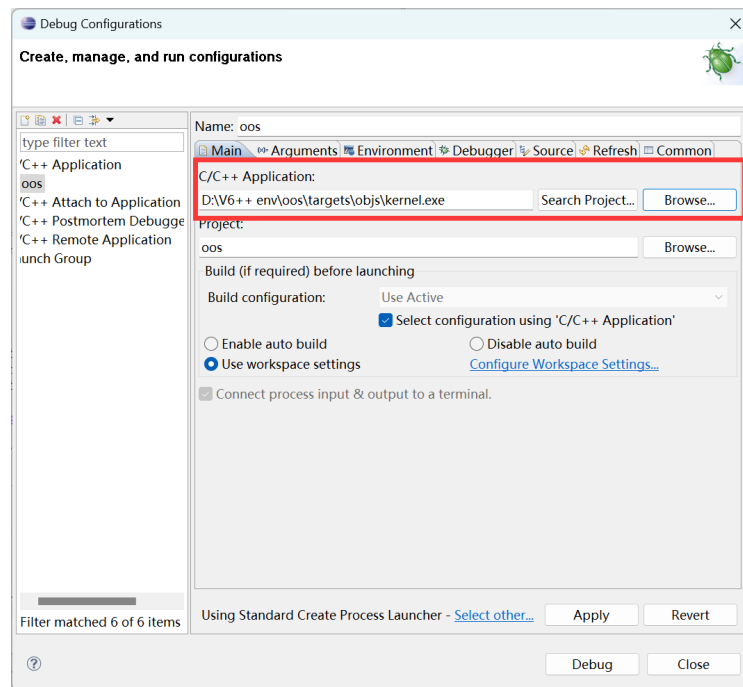
专 业 计算机科学与技术

授课老师 方 钰

P03: UNIX V6++完整的进程图像

一、完成实验 4.1~4.2, 依照实验指导的步骤, 获取进程 User 结构、Proc 结构和 Text 结构的内容, 截图或绘制表格说明, 并总结出在 UNIX V6++中获取进程的代码段和可交换部分起始位置的逻辑地址和物理地址的方法。

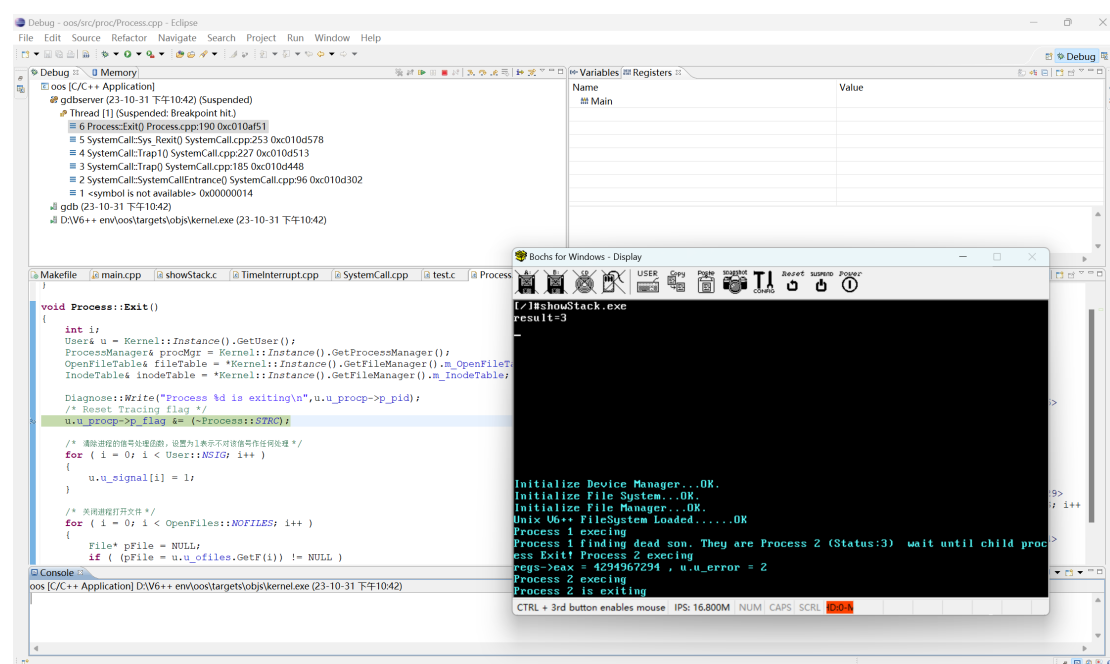
1. 将调试对象设置为 Kernel.exe:



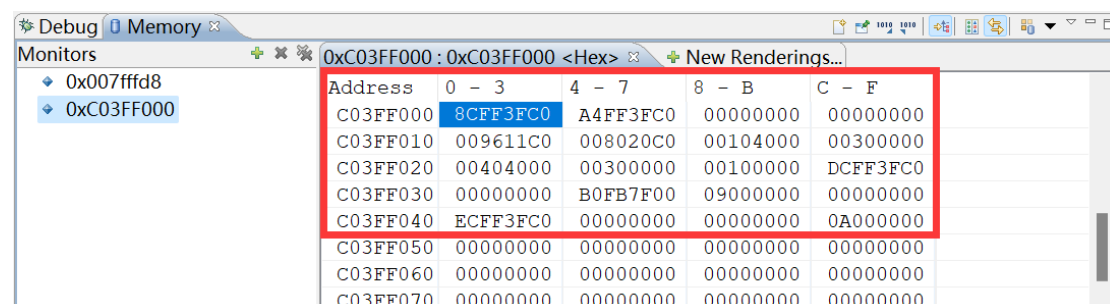
2. 在 Process::Exit()函数中设置断点:



3. 调试过程中，程序执行完输出语句后，停在 Process:Exit()的断点处：



4. 获取进程的 User 结构：



变量名称	含义	值
Process* u_procp	Proc 结构的逻辑地址	0xC0119600
MemoryDescriptor u_MemoryDescriptor (定义如下，此处均为逻辑地址)		
PageTable* m_UserPageTableArray	相对映射表首地址	0xC0208000
unsigned long m_TextStartAddress	代码段起始地址	0x00401000=4M+4K
unsigned long m_TextSize	代码段长度	0x00003000=12K
unsigned long m_DataStartAddress	数据段起始地址	0x00404000=4M+16K
unsigned long m_DataSize	数据段长度	0x00003000=12K
unsigned long m_StackSize	栈段长度	0x00001000=4K

5. 获取进程的 Proc 结构:

Address	0 - 3	4 - 7	8 - B	C - F
C0119600	00000000	02000000	01000000	00F04000
C0119610	00500000	94AE11C0	03000000	01000000
C0119620	65000000	1A000000	00000000	00000000
C0119630	00000000	00000000	A00D12C0	00000000
C0119640	00000000	00000000	FFFFFFFF	00000000
C0119650	00000000	00000000	00000000	00000000
C0119660	00000000	00000000	00000000	00000000
C0119670	00000000	00000000	00000000	00000000
C0119680	00000000	00000000	FFFFFFFF	00000000
C0119690	00000000	00000000	00000000	00000000
C01196A0	00000000	00000000	00000000	00000000
C01196B0	00000000	00000000	00000000	00000000

变量名称	含义	值
<code>short p_uid</code>	用户 ID	0
<code>int p_pid</code>	进程标识数	2
<code>int p_ppid</code>	父进程标识数	1
<code>unsigned long p_addr</code>	user 结构即 ppda 区的物理地址	0x0040F000
<code>unsigned int p_size</code>	除共享正文段的长度, 以字节单位	0x00005000=20K
<code>Text* p_textp</code>	指向代码段 Text 结构的逻辑地址	0xC011AE94
<code>ProcessState p_stat</code>	进程调度状态	3=SRUN
<code>int p_flag</code>	进程标志位	1=SLOAD
<code>int p_pri</code>	进程优先数	0x65
<code>int p_cpu</code>	cpu 值, 用于计算 p_pri	0x1A
<code>int p_nice</code>	进程优先数微调参数	0
<code>int p_time</code>	进程在盘上 (内存内) 驻留时间	0
<code>unsigned long p_wchan</code>	进程睡眠原因	0

6. 获取进程代码段的 Text 结构:

Address	0 - 3	4 - 7	8 - B	C - F
C011AE90	01000100	70460000	00C04000	00300000
C011AEA0	84EC11C0	01000100	00000000	00000000
C011AEB0	00000000	00000000	00000000	00000000
C011AEC0	00000000	00000000	00000000	00000000
C011AED0	00000000	00000000	00000000	00000000
C011AEE0	00000000	00000000	00000000	00000000
C011AEF0	00000000	00000000	00000000	00000000
C011AF00	00000000	00000000	00000000	00000000
C011AF10	00000000	00000000	00000000	00000000
C011AF20	00000000	00000000	00000000	00000000
C011AF30	00000000	00000000	00000000	00000000
C011AF40	00000000	00000000	00000000	00000000

变量名称	含义	值
<code>int x_daddr</code>	代码段在盘交换区上的地址	0x00004670
<code>unsigned long x_caddr</code>	代码段起始地址 (物理地址)	0x0040C000
<code>unsigned int x_size</code>	代码段长度, 以字节为单位	0x00003000 = 12K
<code>Inode* x_iptr</code>	内存 inode 地址	0xC011ECD0
<code>Unsigned short x_count</code>	共享正文段的进程数	1
<code>Unsigned short x_ccount</code>	共享该正文段且图像在内存的进程数	1

7. 进程图象如下表所示:

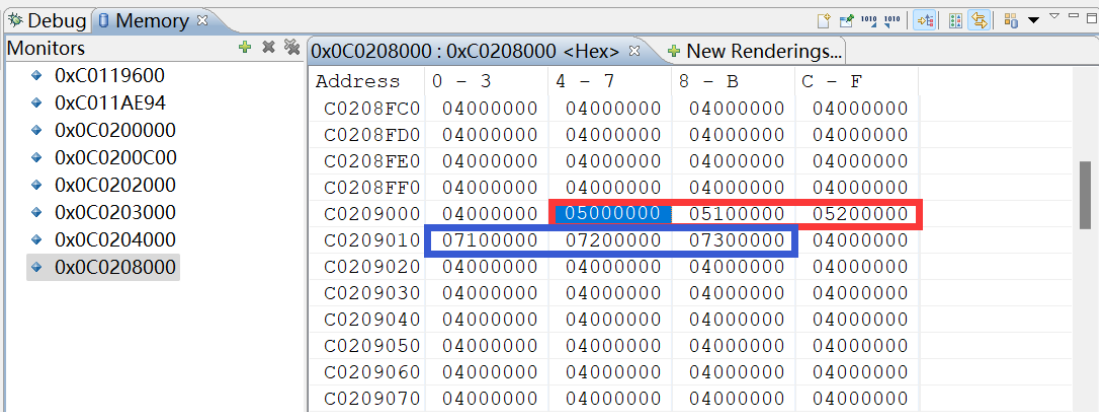
名称	逻辑地址	物理地址	大小
代码段	0x00401000	0x0040C000	12K
可交换部分	0xC03FF000	0x0040F000	20K
PPDA 区	0xC03FF000	0x0040F000	4K
数据段	0x00404000	0x00410000	12K
堆栈段		0x00413000	1K

获取进程的代码段和可交换部分起始位置的逻辑地址和物理地址的方法如下:

1. 可交换部分的逻辑地址, 即 PPDA 区的逻辑地址固定为 0xC03FF000;
2. 可交换部分的物理地址即 `User->u_procp->p_addr`;
3. 代码段的逻辑地址, 即 `User->m_TextStartAddress`;
4. 代码段的物理地址, 即 `User->u_procp->p_textp->x_caddr`。

二、完成实验 4.3, 获取完整的进程相对虚实地址映射表, 补齐表 5。

1. 在 Memory 窗口中查看相对虚实地址映射表:



Address	0 - 3	4 - 7	8 - B	C - F
C0208FC0	04000000	04000000	04000000	04000000
C0208FD0	04000000	04000000	04000000	04000000
C0208FE0	04000000	04000000	04000000	04000000
C0208FF0	04000000	04000000	04000000	04000000
C0209000	04000000	05000000	05100000	05200000
C0209010	07100000	07200000	07300000	04000000
C0209020	04000000	04000000	04000000	04000000
C0209030	04000000	04000000	04000000	04000000
C0209040	04000000	04000000	04000000	04000000
C0209050	04000000	04000000	04000000	04000000
C0209060	04000000	04000000	04000000	04000000
C0209070	04000000	04000000	04000000	04000000

2. 补齐表 5 如下：

页号	地址	值	
		高 20 位页框号	低 12 位标志位 (u/s r/w p)
0#	0xC0208000~0xC0208003		
		
	0xC0209000~0xC0209003		
1024#			
1025#	0xC0209004~0xC0209007	0	005 (0000 0000 0101)
1026#	0xC0209008~0xC020900B	1	005 (0000 0000 0101)
1027#	0xC020900C~0xC020900F	2	005 (0000 0000 0101)
1028#	0xC0209010~0xC0209013	1	007 (0000 0000 0111)
1029#	0xC0209014~0xC0209017	2	007 (0000 0000 0111)
1030#	0xC0209018~0xC020901B	3	007 (0000 0000 0111)
1031#	0xC020901C~0xC020901F		004 (0000 0000 0100)

	0xC0209FFC~0xC0209FFF	4	007 (0000 0000 0111)

三、 首先回答现运行进程的四张页表的逻辑地址是什么， 再根据这个逻辑地址获取完整的进程物理页表， 仿照表 5， 自行绘制表格说明。

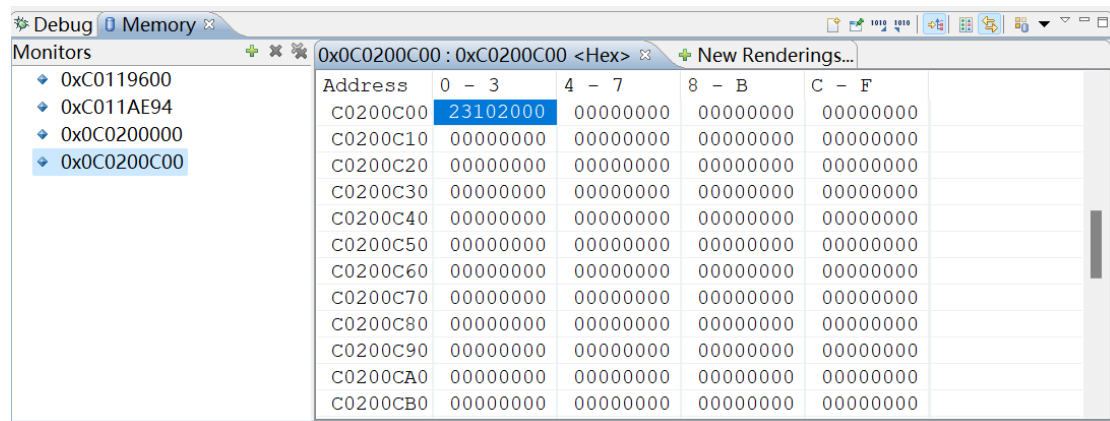
四张页表的逻辑地址为：

- 0xC0200000（页目录），
- 0xC0201000（#768 号页表），
- 0xC0202000（#0 号页表），
- 0xC0203000（#1 号页表）

四张页表的逻辑地址为 3G + 2M 开始的依次每 4K 个字节。可以通过页目录（地址 0xC0200000）观察 0#、1#号页表的逻辑地址（高位部分）：

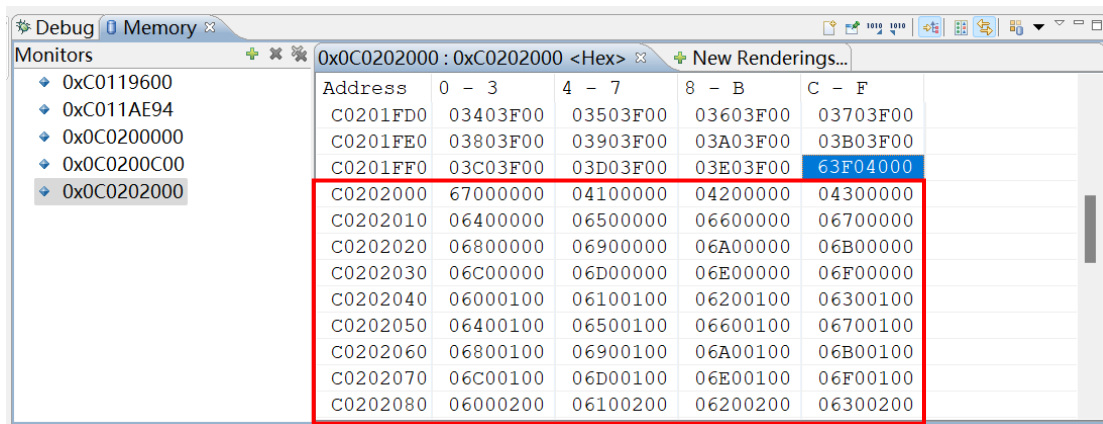
Monitors					
0x0C0200000 : 0xC0200000 <Hex>					
◆ 0xC0119600 ◆ 0xC011AE94 ◆ 0x0C0200000	Address	0 - 3	4 - 7	8 - B	C - F
	C0200000	27202000	27302000	00000000	00000000
	C0200010	00000000	00000000	00000000	00000000
	C0200020	00000000	00000000	00000000	00000000
	C0200030	00000000	00000000	00000000	00000000
	C0200040	00000000	00000000	00000000	00000000
	C0200050	00000000	00000000	00000000	00000000
	C0200060	00000000	00000000	00000000	00000000
	C0200070	00000000	00000000	00000000	00000000
	C0200080	00000000	00000000	00000000	00000000
	C0200090	00000000	00000000	00000000	00000000
	C02000A0	00000000	00000000	00000000	00000000
	C02000B0	00000000	00000000	00000000	00000000

在 $3G + 2M + 768 * 4(\text{Byte})$ 处，可以观察到#768 号页表的逻辑地址（高位部分）：



Address	0 - 3	4 - 7	8 - B	C - F
C0200C00	23102000	00000000	00000000	00000000
C0200C10	00000000	00000000	00000000	00000000
C0200C20	00000000	00000000	00000000	00000000
C0200C30	00000000	00000000	00000000	00000000
C0200C40	00000000	00000000	00000000	00000000
C0200C50	00000000	00000000	00000000	00000000
C0200C60	00000000	00000000	00000000	00000000
C0200C70	00000000	00000000	00000000	00000000
C0200C80	00000000	00000000	00000000	00000000
C0200C90	00000000	00000000	00000000	00000000
C0200CA0	00000000	00000000	00000000	00000000
C0200CB0	00000000	00000000	00000000	00000000

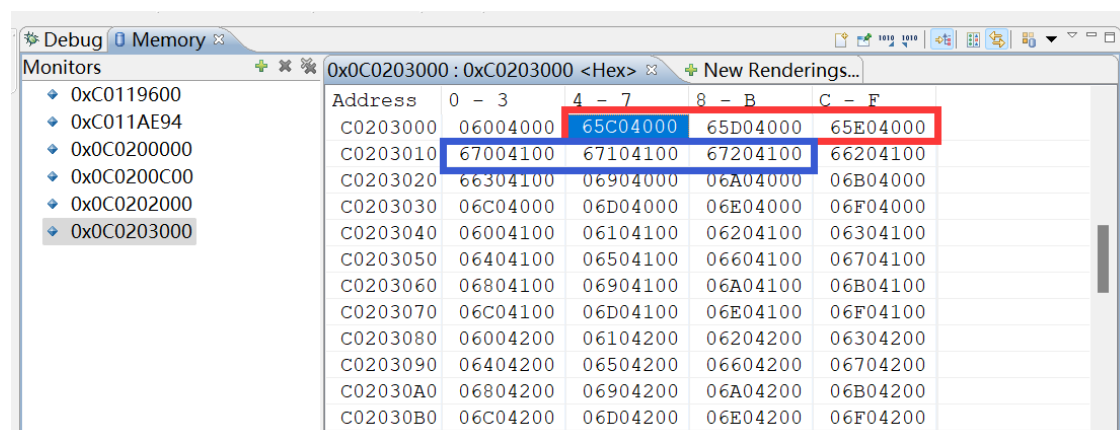
首先获取#768 号页表的信息。查看其最后一个页框，即对应 PPDA 区的地址（下图蓝色）：0x040F000：



Address	0 - 3	4 - 7	8 - B	C - F
C0201FD0	03403F00	03503F00	03603F00	03703F00
C0201FE0	03803F00	03903F00	03A03F00	03B03F00
C0201FF0	03C03F00	03D03F00	03E03F00	63F04000
C0202000	67000000	04100000	04200000	04300000
C0202010	06400000	06500000	06600000	06700000
C0202020	06800000	06900000	06A00000	06B00000
C0202030	06C00000	06D00000	06E00000	06F00000
C0202040	06000100	06100100	06200100	06300100
C0202050	06400100	06500100	06600100	06700100
C0202060	06800100	06900100	06A00100	06B00100
C0202070	06C00100	06D00100	06E00100	06F00100
C0202080	06000200	06100200	06200200	06300200

#0 号页表对应的逻辑地址是 0xC0202000，对应上图中下方的红框处。

#1 号页表对应的逻辑地址是 0xC0203000。可以观察到下图中红框部分对应的为代码段，蓝色部分对应的为数据段。



Address	0 - 3	4 - 7	8 - B	C - F
C0203000	06004000	65C04000	65D04000	65E04000
C0203010	67004100	67104100	67204100	66204100
C0203020	66304100	06904000	06A04000	06B04000
C0203030	06C04000	06D04000	06E04000	06F04000
C0203040	06004100	06104100	06204100	06304100
C0203050	06404100	06504100	06604100	06704100
C0203060	06804100	06904100	06A04100	06B04100
C0203070	06C04100	06D04100	06E04100	06F04100
C0203080	06004200	06104200	06204200	06304200
C0203090	06404200	06504200	06604200	06704200
C02030A0	06804200	06904200	06A04200	06B04200
C02030B0	06C04200	06D04200	06E04200	06F04200

#1 号页表最后一个页框对应的逻辑地址即对应下图蓝色处，其对应堆栈段部分，可以观察到地址与数据段相连：

Monitors		0x0C0204000 : 0xC0204000 <Hex>				New Renderings...	
		Address	0 - 3	4 - 7	8 - B	C - F	
0xC0119600		C0203FD0	06407F00	06507F00	06607F00	06707F00	
0xC011AE94		C0203FE0	06807F00	06907F00	06A07F00	06B07F00	
0x0C0200000		C0203FF0	06C07F00	06D07F00	06E07F00	67304100	
0x0C0200C00		C0204000	00000000	00000000	00000000	00000000	
0x0C0202000		C0204010	00000000	00000000	00000000	00000000	
0x0C0203000		C0204020	00000000	00000000	00000000	00000000	
0x0C0204000		C0204030	00000000	00000000	00000000	00000000	
		C0204040	00000000	00000000	00000000	00000000	
		C0204050	00000000	00000000	00000000	00000000	
		C0204060	00000000	00000000	00000000	00000000	
		C0204070	00000000	00000000	00000000	00000000	
		C0204080	00000000	00000000	00000000	00000000	

页目录：

页号	地址	值	
		高 20 位页框号	低 12 位标志位 (u/s r/w p)
0#	0xC0200000~0xC0200003	0x00202	027 (0000 0010 0111)
1#	0xC0200000~0xC0200007	0x00203	027 (0000 0010 0111)

768#	0xC0200C00~0xC0200C03	0x00201	023 (0000 0010 0111)

768#号页表：

页号	地址	值	
		高 20 位页框号	低 12 位标志位 (u/s r/w p)
0#	0xC0201000~0xC0201003	0x00000	003 (0000 0000 0111)
1#	0xC0201004~0xC0201007	0x00001	003 (0000 0000 0111)
		
1023#	0xC020100C~0xC020100F	0x0040F	063 (0000 0110 0111)

0#号页表：

页号	地址	值	
		高 20 位页框号	低 12 位标志位 (u/s r/w p)

0#	0xC0202000~0xC0202003		
----	-----------------------	--	--

1#号页表:

页号	地址	值	
		高 20 位页框号	低 12 位标志位 (u/s r/w p)
0#	0xC0203000~0xC0203003		
1#	0xC0203004~0xC0203007	0x0040C	065 (0000 0110 0101)
2#	0xC0203008~0xC020300B	0x0040D	065 (0000 0110 0101)
3#	0xC020300C~0xC020300F	0x0040E	065 (0000 0110 0101)
4#	0xC0203010~0xC0203013	0x00410	067 (0000 0110 0111)
5#	0xC0203014~0xC0203017	0x00411	067 (0000 0110 0111)
6#	0xC0203018~0xC020301B	0x00412	067 (0000 0110 0111)

1023#	0xC0203FFC~0xC0203FFF	0x00413	067 (0000 0110 0111)

四、根据实验结果，绘制进程完整的图象，包括进程图象在内存中的位置，相对虚实地址映射表、物理页表在内存的位置和内容，可参考讲义图 4.30。

