

# 윈도우 시스템 구조

---



*proneer*

*proneer@gmail.com*

*<http://forensic-proof.com>*

*Security is a people problem...*



1. 물리메모리 (Physical Memory)
2. 메모리보호 (Memory Protection)
3. 가상메모리 (Virtual Memory)
4. 사용자모드와 커널모드 (User Mode and Kernel Mode)

# Physical Memory

- Physical Address Extension (PAE)
- Data Execution Prevention (DEP)
- Address Windowing Extensions (AWE)
- Pages, Page Frames and Page Frame Numbers



## 소개

- 물리 메모리 확인 방법

- BIOS의 POST 작업에서 확인된 메모리와 비교
- 시스템 등록정보를 통해서도 확인 가능

forensicsight.org

Page 4 / 48



## 물리 주소 확장 (PAE, Physical Address Extension)

- 운영체제 버전, 하드웨어 플랫폼, 구성 방식에 따라 접근 가능한 메모리 크기

Version	Limit for 32-bit Hardware	Limit for 64-bit Hardware
Windows XP Home, Mediacenter	4 GB	N/A
Windows XP Professional	4 GB	128 GB
Windows Server 2003 Standard	4 GB	32 GB
Windows Server 2003 R2 Enterprise, Datacenter	64 GB	1 TB
Windows Vista Business, Enterprise, Ultimate	4 GB	128 GB
Windows Server 2008 Standard, Web	4 GB	32 GB
Windows Server 2008 Enterprise, Datacenter	4 GB	2 TB
Windows 7 Home Premium	4 GB	16 GB
Windows 7 Pro, Enterprise, Ultimate	4 GB	192 GB
Windows Server 2008 R2 Standard	N/A	32 GB
Windows Server 2008 R2 Enterprise	N/A	2 TB



## 물리 주소 확장 (PAE, Physical Address Extension)

### ▪ 물리적 주소 확장

- 물리적 주소 지정 비트를 32비트에서 36비트로 확장 (4GB → 64GB)
- 접근 가능한 물리 주소 공간 증가

### ▪ 윈도우 Vista 이후의 PAE 설정

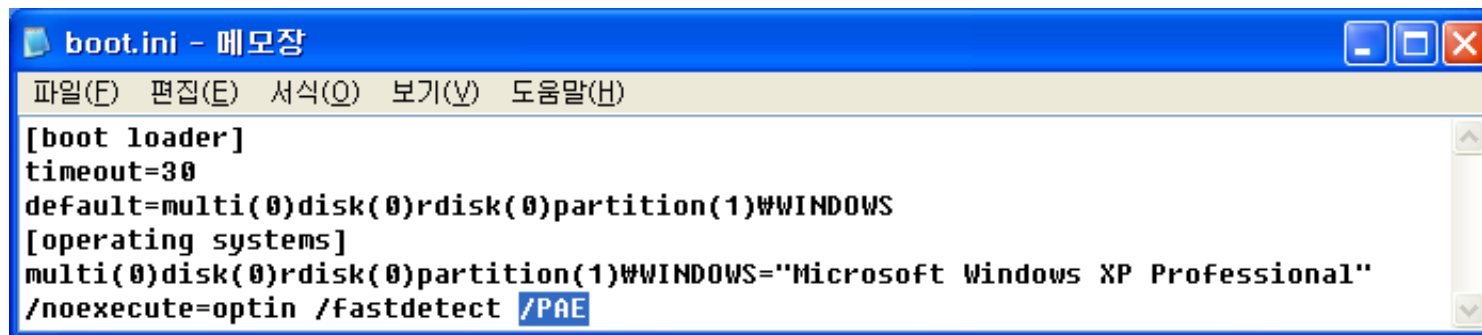
- **bcdedit** (BCD – Boot Configuration Data)

```
관리자: C:\Windows\system32\cmd.exe  
C:\Users\proneer>bcdedit /set PAE ForceEnable  
작업을 완료했습니다.  
C:\Users\proneer>
```



## 물리 주소 확장 (PAE, Physical Address Extension)

- 윈도우 2003 이전에서 PAE 설정



```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Professional"
/noexecute=optin /fastdetect /PAE
```

- boot.ini 파일에 /PAE 스위치 추가
- 증가된 주소 공간에 접근하기 위해서는 AEW(Address Windowing Extensions) API 사용
- PAE로 확장된 공간은 보통 램 디스크(RAM Disk)로 활용



## 데이터 실행 방지 (DEP, Data Execution Prevention)

### ▪ 데이터 실행 방지

- 스택, 데이터 세그먼트, 힙과 같은 메모리 페이지를 실행 불가능하도록 설정
- 버퍼 오버플로우와 같은 공격 방지

하드웨어 강제 DEP 설정 비트	CPU 제조사	설명
NX	AMD	No-eXecute page-protection
XD	Intel	eXecution Disable bit

### ▪ 하드웨어 강제(Hardware-enforced)

- CPU의 NX/XD 비트로 설정하며 OS와 사용자 애플리케이션 모두에서 사용 가능
  - ✓ NX/XD 비트를 지원하는 CPU에서만 동작
- PAE가 동작할 때만 설정 가능(NX/XD 비트를 사용할 경우 자동으로 PAE로 부팅)

### ▪ 소프트웨어 강제(Software-enforced)

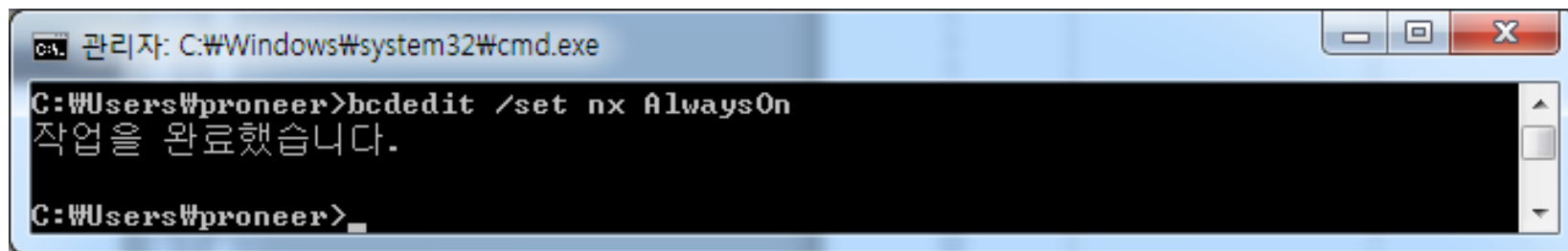
- 비주얼 스튜디오의 /SafeSEH 링커 옵션(SHE, Structured Exception Handler)
- 소프트웨어 예외 처리시 컴파일된 PE 이미지의 예외 처리자에 포함되어 있는지 추가 검사





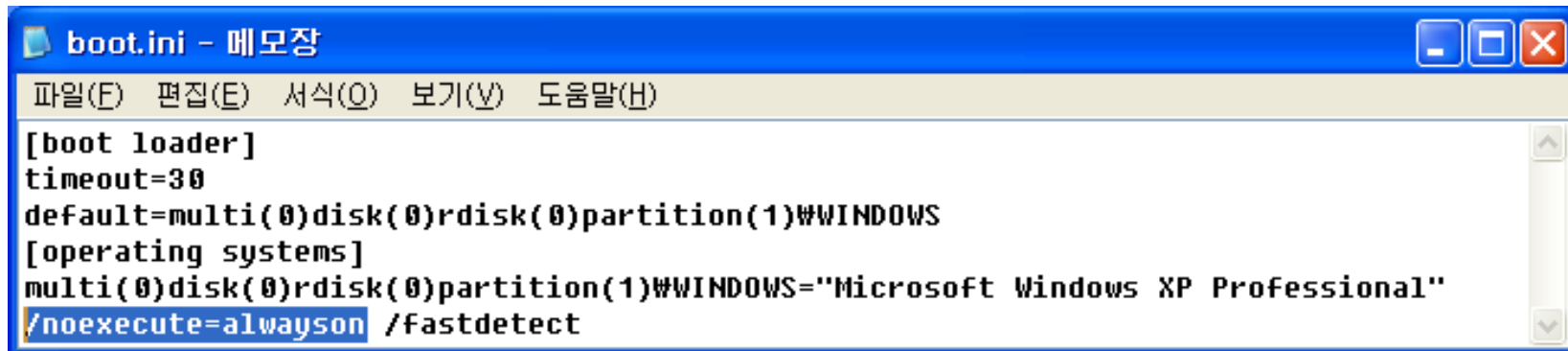
## 데이터 실행 방지 (DEP, Data Execution Prevention)

- 윈도우 Vista/7, 서버 2008에서 하드웨어 강제 DEP 설정
  - bcdedit /set nx AlwaysOn



```
관리자: C:\Windows\system32\cmd.exe
C:\Users\proneer>bcdedit /set nx AlwaysOn
작업을 완료했습니다.
C:\Users\proneer>
```

- 윈도우 서버 2003 이하에서 하드웨어 강제 DEP 설정
  - boot.ini 파일의 /noexecute 스위치 추가

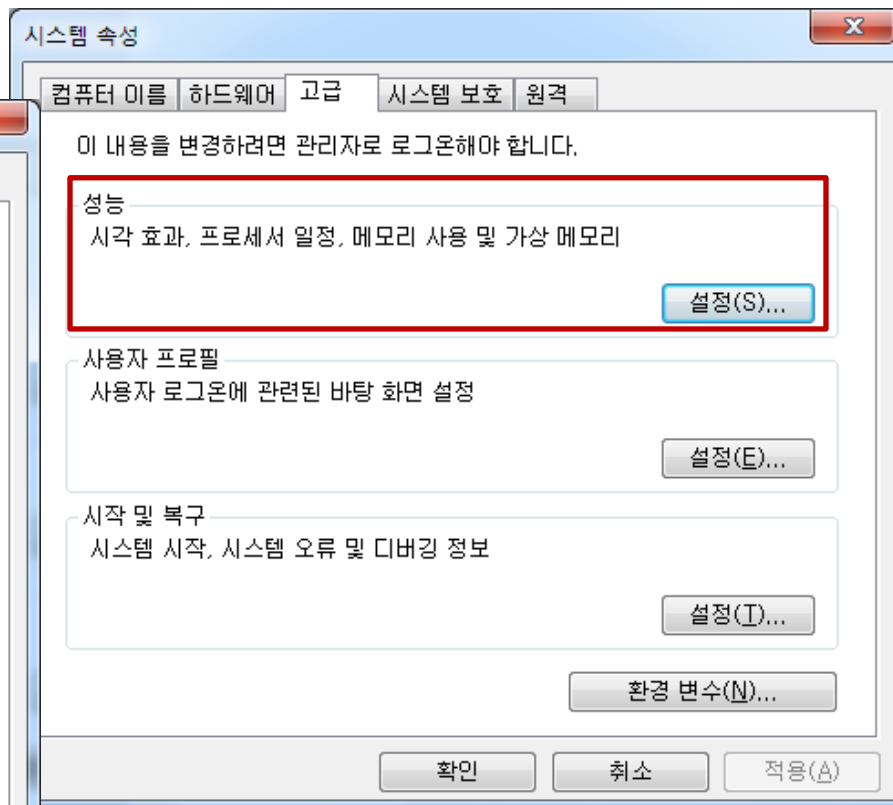
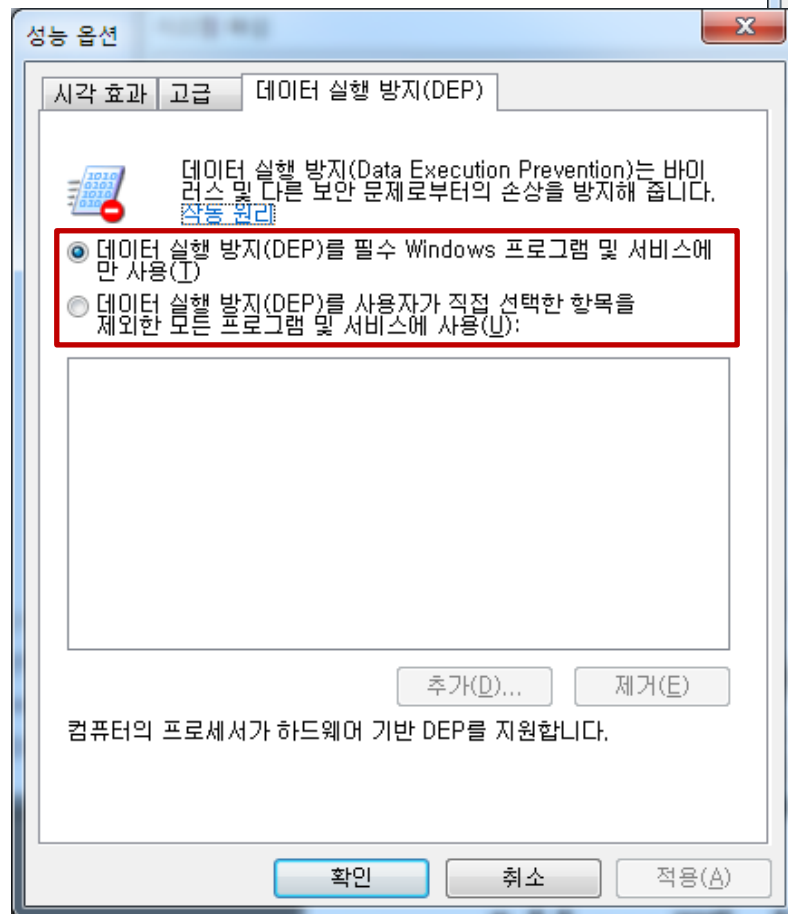


```
boot.ini - 메모장
파일(E) 편집(E) 서식(O) 보기(V) 도움말(H)
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Professional"
/noexecute=alwayson /fastdetect
```



## 데이터 실행 방지 (DEP, Data Execution Prevention)

- 시스템 속성 → 고급 → 성능





## 주소 윈도우 확장 (AWE, Address Windowing Extensions)

- 확장된 메모리 접근 API

- 물리 주소 확장(PAE)에 의해 확장된 메모리를 접근하기 위한 API (winbase.h)

- Lock Pages in Memory

- 사용자 응용프로그램의 경우 가상 메모리로 데이터를 페이징하지 않도록 “Lock Pages in Memory” 가 설정되어야 함

AWE Routine	Description
VirtualAlloc()	Reserves a region in the linear address space of the calling process
VirtualAllocEx()	Reserves a region in the linear address space of the calling process
AllocateUserPhysicalPages()	Allocate pages of physical memory to be mapped to linear memory
MapUserPhysicalPages()	Map allocated pages of physical memory to linear memory
MapUserPhysicalPagesScatter()	Map allocated pages of physical memory to linear memory
FreeUserPhysicalPages()	Release physical memory allocated for use by AWE



## 페이지, 페이지 프레임, 페이지 프레임 번호

### ▪ 페이지(Page)

- 가상 주소 공간에서 연속된 영역
- IA-32 프로세스에서 페이지 크기 : 4KB, 2MB, 4MB (보통 4KB)
- 페이지와 연결된 물리적인 위치는 존재하지 않음
- 페이지는 메모리 또는 디스크 상에 존재

### ▪ 페이지 프레임(Page Frame)

- 페이지가 램이 존재하고 있을 때, 페이지의 물리메모리 상의 위치

### ▪ 페이지 프레임 번호 (PFN, Page Frame Number)

- 페이지 프레임의 물리메모리 위치는 페이지 프레임 번호로 표현



## 페이지, 페이지 프레임, 페이지 프레임 번호

- **20비트 PFN**

- 페이지 크기가 4 KB이고 PAE가 비활성화일 경우, PFN은 20바이트
- 예) 0x12345

- **최하위 12비트는 0**

- 예) 0x12345    0x12345000

- **상위 20바이트 PFN 주소에 항상 페이지 크기를 곱**

# Memory Protection

- Segmentation
- Paging
- Linear to Physical Address Translation

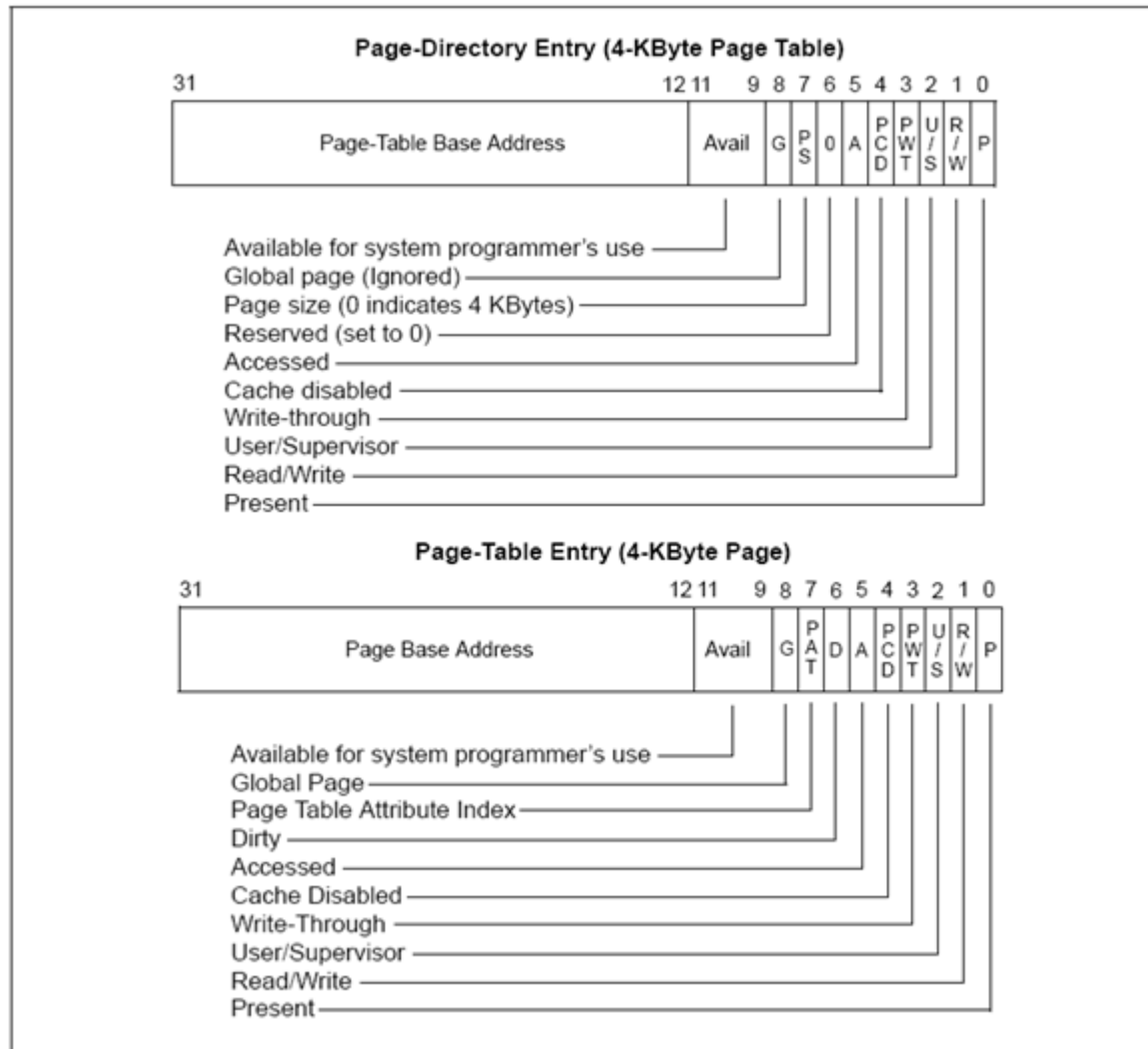


## 메모리 보호 소개

- 운영체제와 사용자 애플리케이션의 경계는 하드웨어 기반 매커니즘에 의존
- IA-32 프로세스의 메모리 보호 기법
  - 페이징 > 세그먼트
- 4개의 링 모델 → 2개의 링 모델
  - 관리자 레벨(Supervisor Level) – 커널 모드
  - 사용자 레벨(User Level) – 사용자 모드
- U/S 비트
  - PDE, PTE에서 커널/사용자 모드의 구분은 U/S 비트를 통해 이루어짐



## 메모리 보호 소개







## 세그멘테이션 (Segmentation)

- 세그먼트는 GDT(Global Descriptor Table)에 정의
- GDTR 레지스터 : GDT의 베이스 주소와 크기 저장
- rM 0x100 → descriptor register 확인

```
kd> rM 0x100
gdtr=8003f000   gdtl=03ff idtr=8003f400   idtl=07ff tr=0028   ldtr=0000
LiveKdD+0x32fd:
f8a852fd eb30                jmp     LiveKdD+0x332f <f8a8532f>
```

```
kd> r gdtr
gdtr=8003f000
kd> r gdtl
gdtl=000003ff
```

- GDT 시작 주소 : 0x8003f000
- GDT 크기 : 1,023 바이트(0x3FF)



## 세그멘테이션 (Segmentation)

- GDT 내용 확인

```
kd> d 0x8003f000 L3FF
8003f000  00 00 00 00 00 00 00 00 00-ff ff 00 00 00 9b cf 00 .....
8003f010  ff ff 00 00 00 93 cf 00-ff ff 00 00 00 fb cf 00 .....
8003f020  ff ff 00 00 00 f3 cf 00-ab 20 00 20 04 8b 00 80 .....
8003f030  01 00 00 f0 df 93 c0 ff-ff 0f 00 f0 fd f3 40 7f .....e.
8003f040  ff ff 00 04 00 f2 00 00-00 00 00 00 00 00 00 00 .....
8003f050  68 00 80 bf 54 89 00 80-68 00 e8 bf 54 89 00 80 h...T...h...T...
8003f060  ff ff 40 2f 02 93 00 00-ff 3f 00 80 0b 92 00 00 ..e/.....?.....
8003f070  ff 03 00 70 ff 92 00 ff-ff ff 00 00 40 9a 00 80 ...p.....e...
8003f080  ff ff 00 00 40 92 00 80-00 00 00 00 00 92 00 00 ....e.....
8003f090  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
8003f0a0  68 00 b8 c8 5e 89 00 82-00 00 00 00 00 00 00 h...^.....
8003f0b0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
8003f0c0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
8003f0d0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
8003f0e0  ff ff 00 a0 6f 9f 00 f8-ff ff 00 00 00 92 00 00 ....o.....
8003f0f0  b7 03 98 c6 4f 98 00 80-ff ff 00 00 00 92 00 00 ....0.....
8003f100  ff ff 00 94 3a 93 40 f8-ff ff 00 94 3a 93 40 f8 ....:..e....:..e.
8003f110  ff ff 00 94 3a 93 40 f8-20 f1 03 80 00 00 00 00 ....:..e. ....
8003f120  28 f1 03 80 00 00 00 00-30 f1 03 80 00 00 00 00 <.....0.....
8003f130  38 f1 03 80 00 00 00 00-40 f1 03 80 00 00 00 00 8.....e.....
```



## 세그멘테이션 (Segmentation)

- GDT 내용 확인

```
kd> dg 0 3f8
```

Sel	Base	Limit	Type	P	Si	Gr	Pr	Lo	Flags
0000	00000000	00000000	<Reserved>	0	Nb	By	Np	Nl	00000000
0008	00000000	fffffff	Code RE Ac	0	Bg	Pg	P	Nl	00000c9b
0010	00000000	fffffff	Data RW Ac	0	Bg	Pg	P	Nl	00000c93
0018	00000000	fffffff	Code RE Ac	3	Bg	Pg	P	Nl	00000cfb
0020	00000000	fffffff	Data RW Ac	3	Bg	Pg	P	Nl	00000cf3
0028	80042000	000020ab	TSS32 Busy	0	Nb	By	P	Nl	0000008b
0030	ffdf000	00001fff	Data RW Ac	0	Bg	Pg	P	Nl	00000c93
0038	7ffdf000	00000fff	Data RW Ac	3	Bg	By	P	Nl	000004f3
0040	00000400	0000ffff	Data RW	3	Nb	By	P	Nl	000000f2
0048	00000000	00000000	<Reserved>	0	Nb	By	Np	Nl	00000000
0050	8054bf80	00000068	TSS32 Avl	0	Nb	By	P	Nl	00000089
0058	8054bfe8	00000068	TSS32 Avl	0	Nb	By	P	Nl	00000089
0060	00022f40	0000ffff	Data RW Ac	0	Nb	By	P	Nl	00000093
0068	000b8000	00003fff	Data RW	0	Nb	By	P	Nl	00000092
0070	ffff7000	000003ff	Data RW	0	Nb	By	P	Nl	00000092
0078	80400000	0000ffff	Code RE	0	Nb	By	P	Nl	0000009a
0080	80400000	0000ffff	Data RW	0	Nb	By	P	Nl	00000092
0088	00000000	00000000	Data RW	0	Nb	By	P	Nl	00000092
0090	00000000	00000000	<Reserved>	0	Nb	By	Np	Nl	00000000
0098	00000000	00000000	<Reserved>	0	Nb	By	Np	Nl	00000000
00A0	825ec8b8	00000068	TSS32 Avl	0	Nb	By	P	Nl	00000089
00A8	00000000	00000000	<Reserved>	0	Nb	By	Np	Nl	00000000
00B0	00000000	00000000	<Reserved>	0	Nb	By	Np	Nl	00000000



## 세그멘테이션 (Segmentation)

- GDT 내용 확인

```
kd> dg 0 3f8
```

Sel	Base	Limit	Type	P	Si	Gr	Pr	Lo	Flags
0000	00000000	00000000	<Reserved>	0	Nb	By	Np	Nl	00000000
0008	00000000	ffffffff	Code RE Ac	0	Bg	Pg	P	Nl	00000c9b
0010	00000000	ffffffff	Data RW Ac	0	Bg	Pg	P	Nl	00000c93
0018	00000000	ffffffff	Code RE Ac	3	Bg	Pg	P	Nl	00000cfb
0020	00000000	ffffffff	Data RW Ac	3	Bg	Pg	P	Nl	00000cf3
0028	80042000	000020ab	TSS32 Busy	0	Nb	By	P	Nl	0000008b
0030	ffdf000	00001fff	Data RW Ac	0	Bg	Pg	P	Nl	00000c93
0038	7ffdf000	00000fff	Data RW Ac	3	Bg	By	P	Nl	000004f3
0040	00000400	0000ffff	Data RW	3	Nb	By	P	Nl	000000f2
0048	00000000	00000000	<Reserved>	0	Nb	By	Np	Nl	00000000
0050	8054bf80	00000068	TSS32 Avl	0	Nb	By	P	Nl	00000089
0058	8054bfe8	00000068	TSS32 Avl	0	Nb	By	P	Nl	00000089
0060	00022f40	0000ffff	Data RW Ac	0	Nb	By	P	Nl	00000093
0068	000b8000	00003fff	Data RW	0	Nb	By	P	Nl	00000092
0070	ffff7000	000003ff	Data RW	0	Nb	By	P	Nl	00000092
0078	80400000	0000ffff	Code RE	0	Nb	By	P	Nl	0000009a
0080	80400000	0000ffff	Data RW	0	Nb	By	P	Nl	00000092
0088	00000000	00000000	Data RW	0	Nb	By	P	Nl	00000092
0090	00000000	00000000	<Reserved>	0	Nb	By	Np	Nl	00000000
0098	00000000	00000000	<Reserved>	0	Nb	By	Np	Nl	00000000
00A0	825ec8b8	00000068	TSS32 Avl	0	Nb	By	P	Nl	00000089
00A8	00000000	00000000	<Reserved>	0	Nb	By	Np	Nl	00000000
00B0	00000000	00000000	<Reserved>	0	Nb	By	Np	Nl	00000000



## 페이징 (Paging)

- 각각의 프로세스는 고유한 CR3 레지스터 값이 할당
- CR3 레지스터
  - 페이지 디렉터리의 PFN 저장 → 프로세스는 고유한 페이지 디렉터리를 가짐
  - KPROCESS의 DirectoryTableBase에 값 저장
  - 커널에 의해 작업 전환 시 수행할 프로세스의 CR3 레지스터가 로드

```
kd> !process 0 0
**** NT ACTIVE PROCESS DUMP ****
PROCESS 825b97c0 SessionId: none Cid: 0004 Peb: 00000000 ParentCid: 0000
DirBase: 00b18000 ObjectTable: e1001cb0 HandleCount: 598.
Image: System

PROCESS 8227bda0 SessionId: none Cid: 0228 Peb: 7ffd4000 ParentCid: 0004
DirBase: 0a400020 ObjectTable: e13c4bf8 HandleCount: 19.
Image: smss.exe
```

- **Cid**(PID), **PEB**(Process Environment Block), **ParentCid**(PPID), **DirBase**(DirectoryTableBase)



## 페이징 (Paging)

- CR3 레지스터

```
kd> !process 0 0
**** NT ACTIVE PROCESS DUMP ****
PROCESS 825b97c0 SessionId: none Cid: 0004 Peb: 00000000 ParentCid: 0000
DirBase: 00b18000 ObjectTable: e1001cb0 HandleCount: 598.
Image: System
```

```
kd> dt nt!_EPROCESS 825B97C0
+0x000 Pcb : _KPROCESS
+0x06c ProcessLock : _EX_PUSH_LOCK
+0x070 CreateTime : _LARGE_INTEGER 0x0
+0x078 ExitTime : _LARGE_INTEGER 0x0
+0x080 RundownProtect : _EX_RUNDOWN_REF
+0x084 UniqueProcessId : 0x00000004 Void
+0x088 ActiveProcessLinks : _LIST_ENTRY [ 0x8227be28 - 0x8055c1d8 ]
```

```
kd> dt nt!_KPROCESS 825B97C0
+0x000 Header : _DISPATCHER_HEADER
+0x010 ProfileListHead : LIST_ENTRY [ 0x825b97d0 - 0x825b97d0 ]
+0x018 DirectoryTableBase : [2] 0xb18000
+0x020 LdtDescriptor : _KGDTENTRY
+0x028 Int21Descriptor : _KIDTENTRY
+0x030 IopmOffset : 0x20ac
+0x032 Iopl : 0 ''
```



## 페이징 (Paging)

### ▪ !pte

- 특정 가상 주소(VA)와 연관된 PDE, PTE 값 확인

```
kd> !pte 30001
                                VA 00030001
PDE at C0600000                PTE at C0000180
contains 0000000015744067      contains 0000000000000000
pfn 15744      ---DA--UWEV    not valid
```

- VA(Virtual Address) : 0x00030001
- PDE의 VA : 0xC0600000                      PDE의 내용(Hex) : 0x0000000015744067
- PTE의 VA : 0xC0000180                      PTE의 내용 (Hex) : 0x0000000000000000
- PDE의 PFN : 15744000
- PDE의 최하위 10비트 디코딩 : ---DA--UWEV



## 페이징 (Paging)

### ▪ Flag Code

Bit	Bit Set	Bit Clear	Description (when bit is set)
0	V	-	Page/Page table is valid (present in memory)
1	W	R	Page/Page table writable (as opposed to being read-only)
2	U	K	Owner is user (as opposed to being owned by the kernel)
3	T	-	Write-through caching is enabled for this Page/Page table
4	N	-	Page/Page table caching is disabled
5	A	-	Page/Page table has been accessed (read from or written to)
6	D	-	Page is dirty (has been written to)
7	L	-	Page is larger than 4 KB (4 MB, or 2 MB if PAE is enabled)
8	G	-	Indicates a global page {related to translation lookaside buffers}
9	C	-	Copy on write is enabled
	E	-	Page contains executable code





## 페이징 (Paging)

- PDE, PTE 내용 확인

```
kd> !pte 0
                                VA 00000000
PDE at C0600000                PTE at C0000000
contains 0000000015744067      contains 0000000000000000
pfn 15744      ---DA---UWEU    not valid

kd> !pte 500000
                                VA 00500000
PDE at C0600010                PTE at C0002800
contains 000000001C1BB067      contains 8000000000BFB2067
pfn 1c1bb      ---DA---UWEU    pfn bfb2      ---DA---UW-U

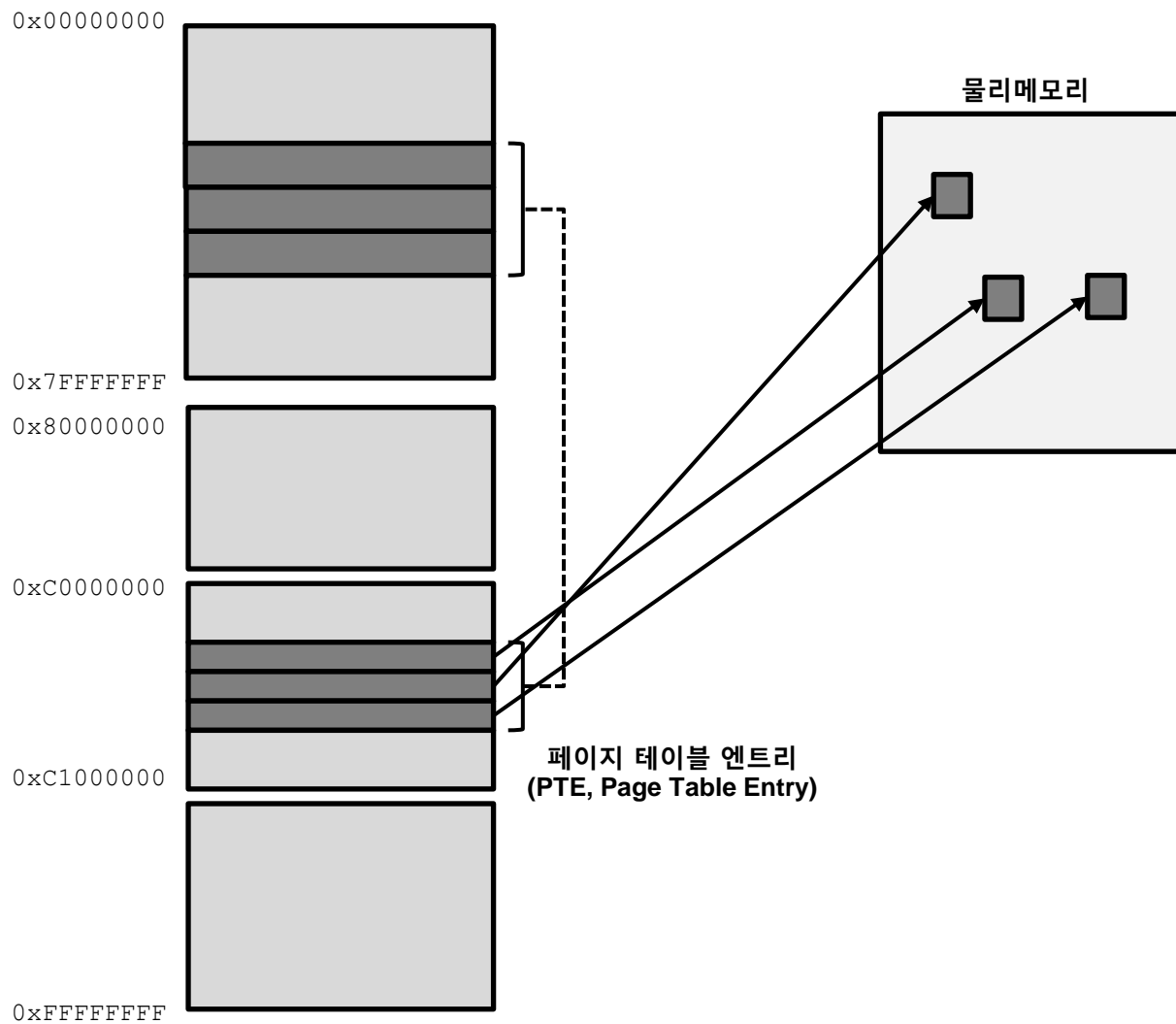
kd> !pte 7fffffff
                                VA 7fffffff
PDE at C0601FF8                PTE at C03FFFF8
contains 000000001EFC1067      contains 0000000000000000
pfn 1efc1      ---DA---UWEU    not valid

kd> !pte 80000000
                                VA 80000000
PDE at C0602000                PTE at C0400000
contains 0000000000B25163      contains 0000000000000000
pfn b25      -G-DA---KWEU    not valid

kd> !pte ffffffff
                                VA ffffffff
PDE at C0603FF8                PTE at C07FFFF8
contains 0000000000B2D163      contains 0000000000000000
pfn b2d      -G-DA---KWEU    not valid
```

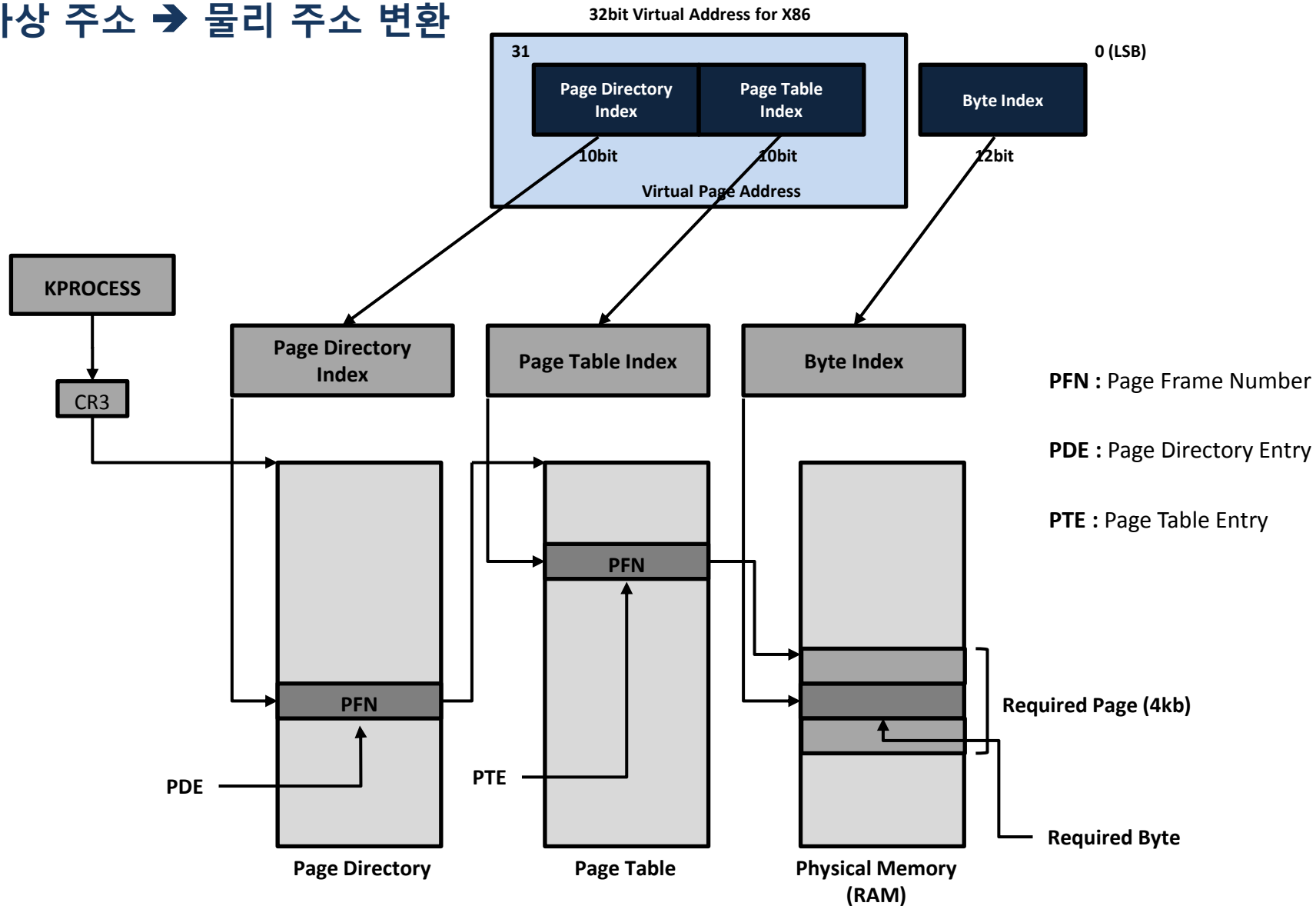


## x86 물리메모리 맵핑





## 가상 주소 → 물리 주소 변환





## 가상 주소 → 물리 주소 변환

### ■ 천천히

- **VA** : 0x00A80910
- **Page Directory Index** : 00000000 10 (0x2)
- **Page Table Index** : 101000 0000 (0x2D0)
- **Page Offset** : 1001 00010000 (0x910)
- **PTE linear address** = (page table starting address) +  
(page directory index) \* (bytes per page table) +  
(page table index) \* (bytes per PTE)  
= (0xC0000000) + (0x2 \* 0x1000) + (0x2D0 \* 0x4)  
= 0xC0002B40

```
kd> dd c0001700
c0001700 0fe96025 00000000 00000000 00000000
```

- **PFN** : 0x0FE96000      **Physical address** : 0x0FE96000 + 0x900 = 0x0FE96900

```
kd> .formats a80910
Evaluate expression:
Hex:      00a80910
Decimal:  11012368
Octal:    00052004420
Binary:   00000000 10101000 00001001 00010000
Chars:    ....
Time:     Fri May 08 19:59:28 1970
Float:    low 1.54316e-038 high 0
Double:   5.44083e-317
```



## 가상 주소 → 물리 주소 변환

### ▪ !PTE 명령 이용

```
kd> !pte a80910
                VA 00a80910
PDE at 00000000C0600028 PTE at 00000000C0005400
contains 00000000CA43067 contains 00000000C9DB067
pfn ca43      ---DA--UWEV      pfn c9db      ---DA--UWEV
```

- PFN : 0xC9DB                      Offset : 0x910
- Physical Address : 0xC9DB910

### ▪ CR3 레지스터 이용

```
kd> r cr3
cr3=086401c0
kd> !vtop 086401c0 a80910
X86VtoP: Virt 00a80910, pagedir 86401c0
X86VtoP: PAE PDPE 86401c0 - 00000000c981001
X86VtoP: PAE PDE c981028 - 00000000ca43067
X86VtoP: PAE PTE ca43400 - 00000000c9db067
X86VtoP: PAE Mapped phys c9db910
Virtual address a80910 translates to physical address c9db910.
```

# Virtual Memory

- User Space Topography
- Kernel Space Dynamic Allocation
- Address Space Layout Randomization (ASLR)

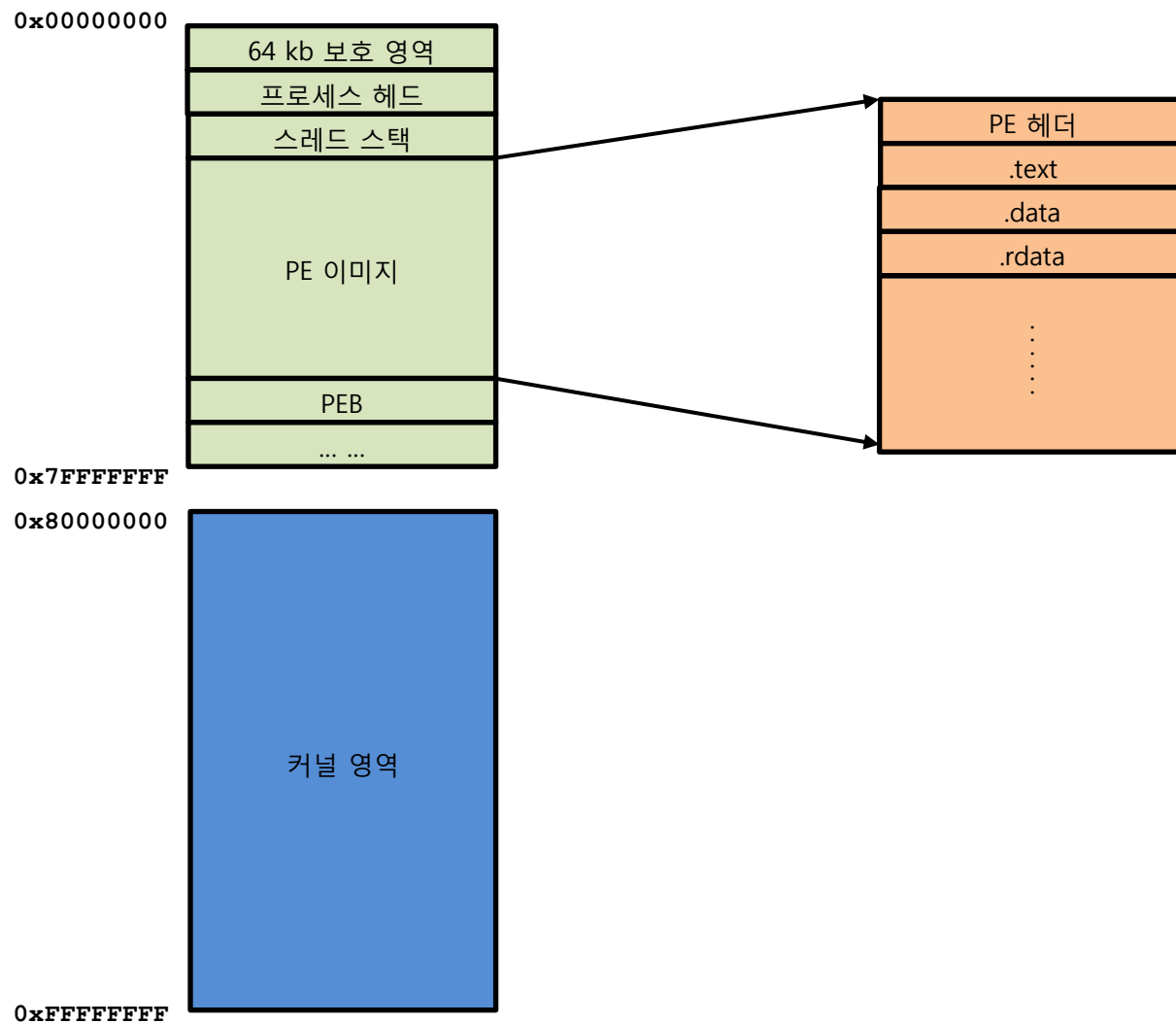


## 가상메모리 소개

- 각 프로세스는 4GB의 가상 주소 공간과 고유한 CR3 레지스터를 가짐
  - 가상 주소 공간의 구분
    - 사용자 영역 (0x00000000 – 0x7FFFFFFF)
    - 커널 영역 (0x80000000 – 0xFFFFFFFF)
  - 사용자 영역 증가
    - 윈도우 Vista 이상
      - ✓ bcdedit /set increaseuserva 3072
- ```
C:\>bcdedit /set increaseuserva 3072
작업을 완료했습니다.
```
- 윈도우 XP, 2003
    - ✓ boot.ini 파일에 /3GB 스위치 추가



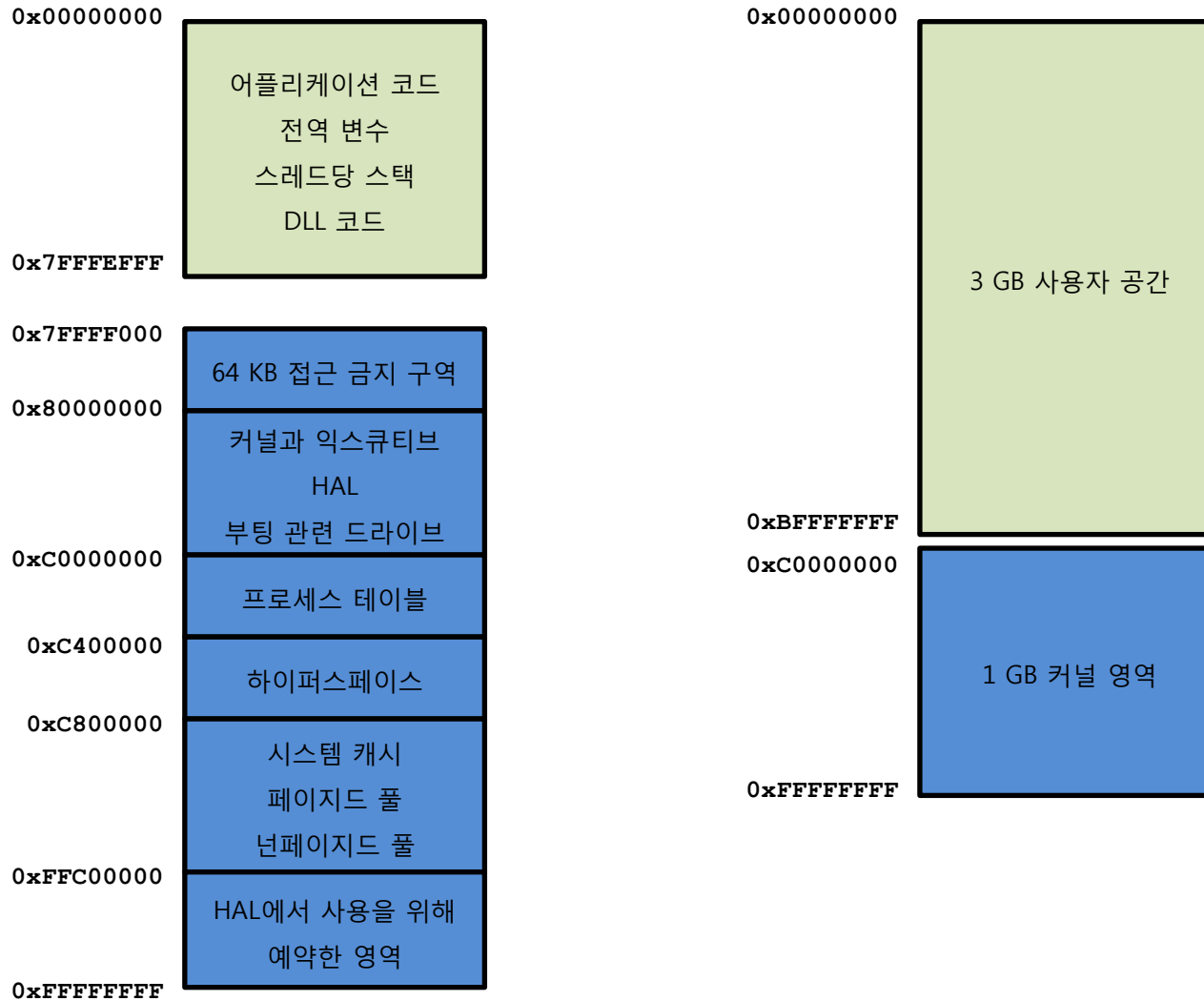
## X86 주소 공간 배치





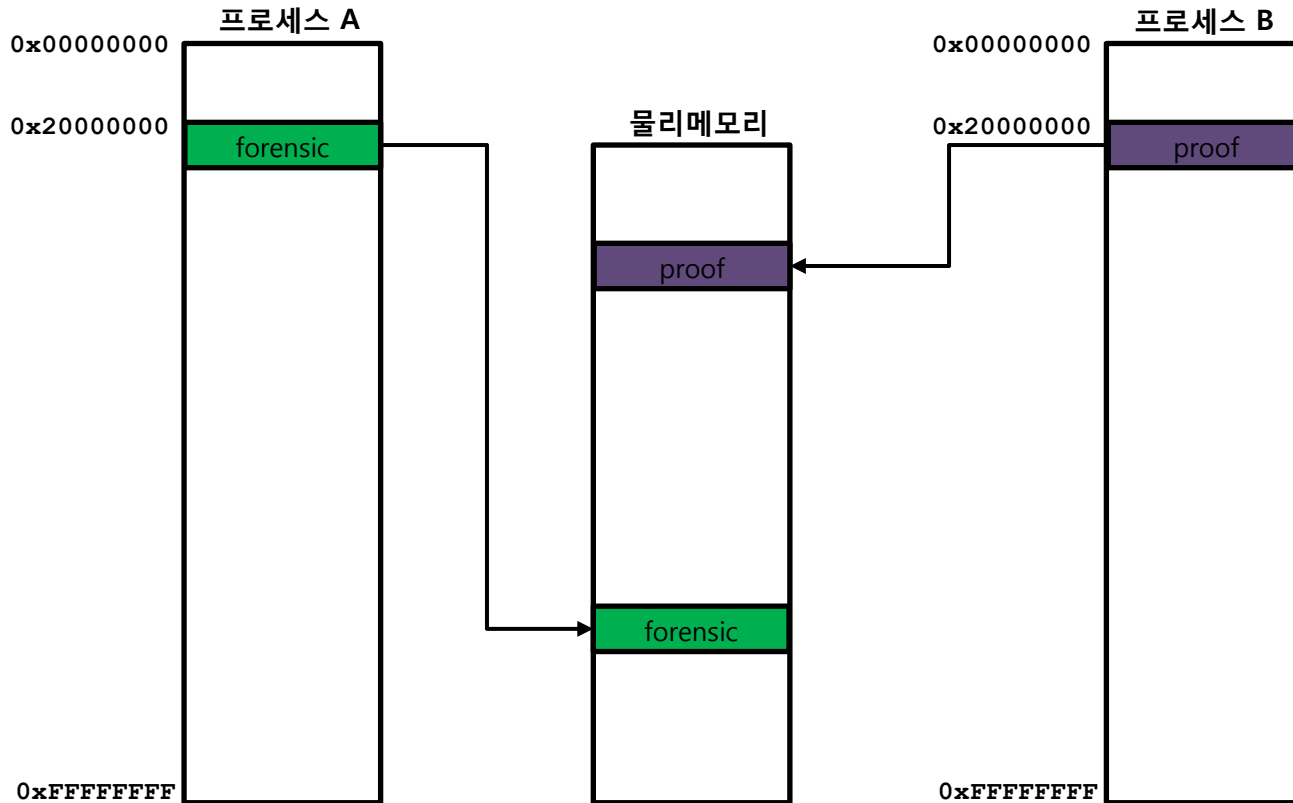


## X86 주소 공간 배치



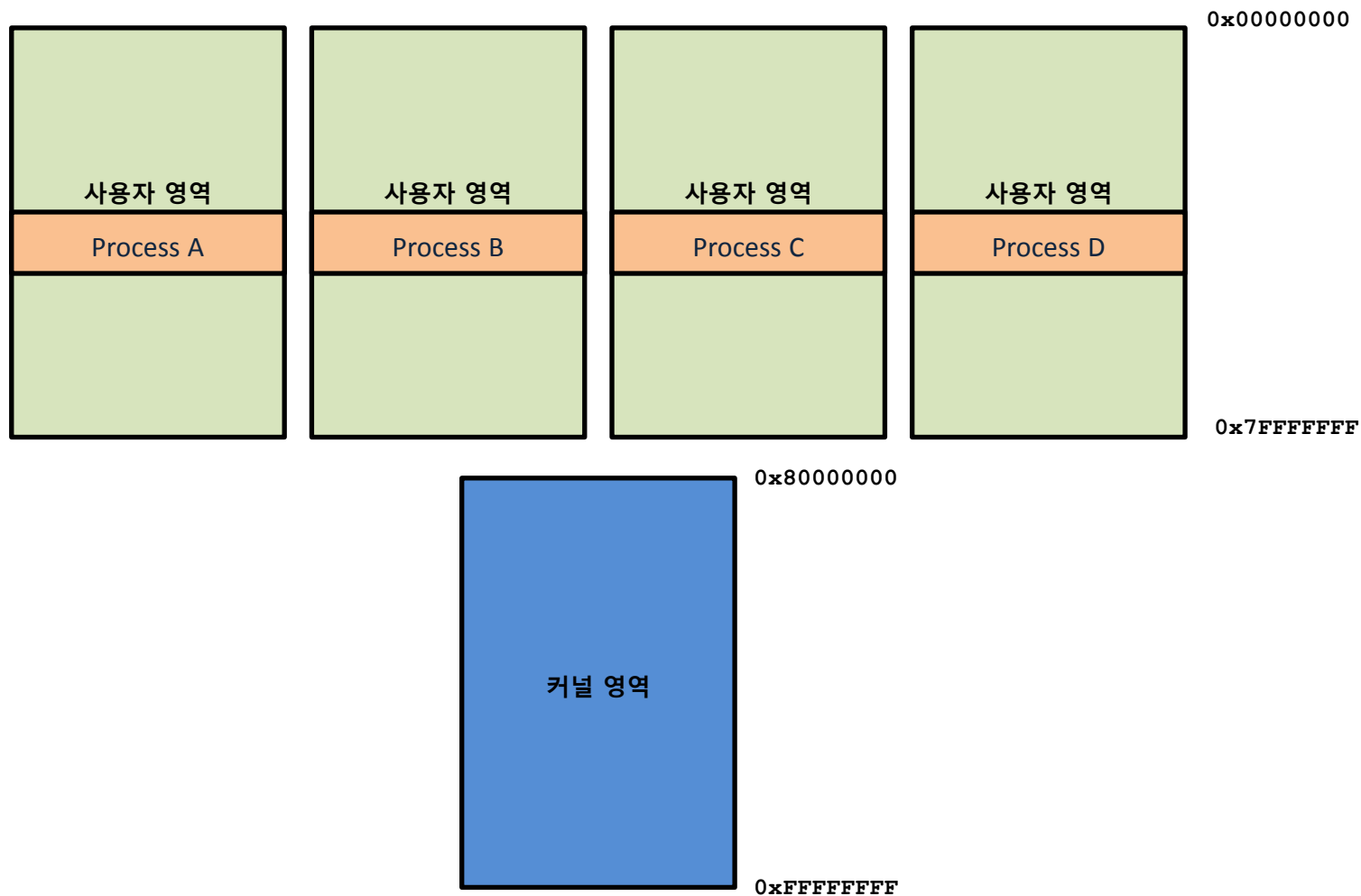


## X86 주소 공간



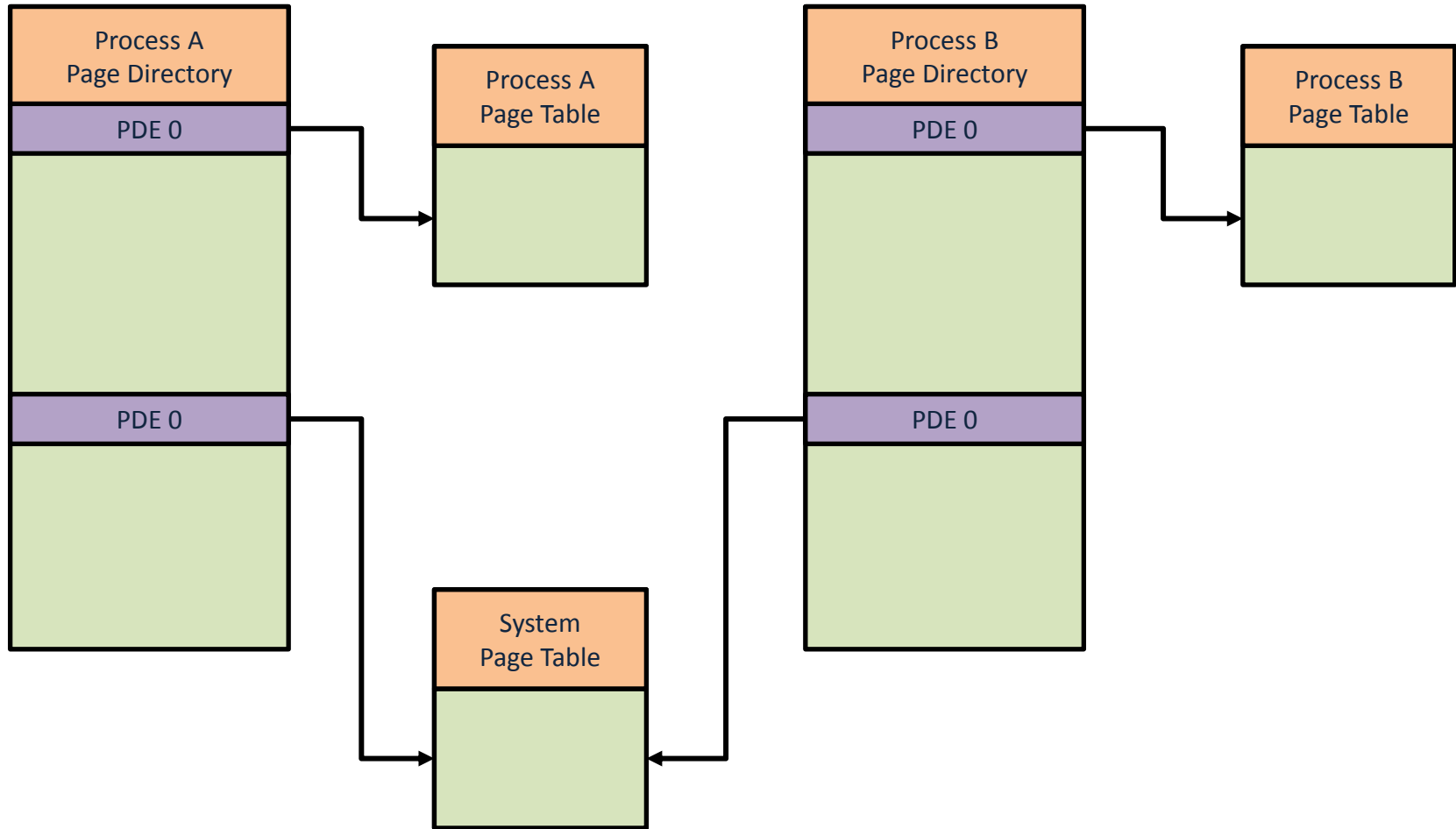


## X86 주소 공간





## X86 주소 공간





## 사용자 공간 구성 (User Space Topography)

```

kd> !process 0 0 explorer.exe
PROCESS 81e5d980 SessionId: 0 Cid: 05cc Peb: 7ffde000 ParentCid: 05b8
DirBase: 086801c0 ObjectTable: e1e556c8 HandleCount: 276.
Image: explorer.exe

kd> .process 81e5d980
Implicit process is now 81e5d980
kd> !peb
PEB at 7ffde000
InheritedAddressSpace: No
ReadImageFileExecOptions: No
BeingDebugged: No
ImageBaseAddress: 01000000
Ldr 00191e90
Ldr.Initialized: Yes
Ldr.InInitializationOrderModuleList: 00191f28 . 00194908
Ldr.InLoadOrderModuleList: 00191ec0 . 001948f8
Ldr.InMemoryOrderModuleList: 00191ec8 . 00194900

```

| Base     | TimeStamp                     | Module                           |
|----------|-------------------------------|----------------------------------|
| 10000000 | 48025c30 Apr 14 04:17:04 2008 | C:\WINDOWS\Explorer.EXE          |
| 7c900000 | 4802a12c Apr 14 09:11:24 2008 | C:\WINDOWS\system32\ntdll.dll    |
| 7c800000 | 4802a12c Apr 14 09:11:24 2008 | C:\WINDOWS\system32\kernel32.dll |
| 77dd0000 | 4802a0b2 Apr 14 09:09:22 2008 | C:\WINDOWS\system32\ADVAPI32.dll |
| 77e70000 | 4802a106 Apr 14 09:10:46 2008 | C:\WINDOWS\system32\RPCRT4.dll   |
| 77fe0000 | 4802a11b Apr 14 09:11:07 2008 | C:\WINDOWS\system32\Secur32.dll  |
| 75f80000 | 4802a0a8 Apr 14 09:09:12 2008 | C:\WINDOWS\system32\BROWSEUI.dll |
| 77f10000 | 4802a0be Apr 14 09:09:34 2008 | C:\WINDOWS\system32\GDI32.dll    |
| 7e410000 | 4802a11b Apr 14 09:11:07 2008 | C:\WINDOWS\system32\USER32.dll   |
| 77c10000 | 4802a188 Apr 14 09:12:56 2008 | C:\WINDOWS\system32\msvcrt.dll   |
| 774e0000 | 4802a111 Apr 14 09:10:57 2008 | C:\WINDOWS\system32\ole32.dll    |
| 77f60000 | 4802a116 Apr 14 09:11:02 2008 | C:\WINDOWS\system32\SHLWAPI.dll  |
| 77120000 | 4802a112 Apr 14 09:10:58 2008 | C:\WINDOWS\system32\OLEAUT32.dll |
| 7e290000 | 4802a110 Apr 14 09:10:56 2008 | C:\WINDOWS\system32\SHDOCVW.dll  |
| 77a80000 | 4802a0d7 Apr 14 09:09:59 2008 | C:\WINDOWS\system32\CRYPT32.dll  |
| 77b20000 | 4802a126 Apr 14 09:11:18 2008 | C:\WINDOWS\system32\MSASN1.dll   |
| 754d0000 | 4802a0dd Apr 14 09:10:05 2008 | C:\WINDOWS\system32\CRYPTUI.dll  |



## 사용자 공간 구성 (User Space Topography)

- 윈도우 Vista 이후 커널은 동적 할당 기능을 사용 (하드코딩 루트킷 방지)

```
0: kd> lm n
start      end          module name
80bab000 80bb3000    kdcom      kdcom.dll
82e10000 83222000    nt         ntkrnpamp.exe
83222000 83259000    hal        halmacpi.dll
8c200000 8c218000    rasl2tp    rasl2tp.sys
8c229000 8c2ae000    mcupdate_GenuineIntel mcupdate_GenuineIntel
8c2ae000 8c2bf000    PSHEd      PSHEd.dll
8c2bf000 8c2c7000    BOOTVID    BOOTVID.dll
8c2c7000 8c309000    CLFS        CLFS.SYS
8c309000 8c3b4000    CI          CI.dll
8c3b4000 8c3d7000    ataport     ataport.SYS
8c3e8000 8c3f9000    termdd      termdd.sys
8c400000 8c409000    amdxtata    amdxtata.sys
8c40b000 8c47c000    Wdf01000    Wdf01000.sys
8c47c000 8c48a000    WDFLDR      WDFLDR.SYS
8c48a000 8c4d2000    ACPI        ACPI.sys
8c4d2000 8c4db000    WMILIB      WMILIB.SYS
8c4db000 8c4e3000    msisadrv     msisadrv.sys
8c4e3000 8c50d000    pci          pci.sys
8c50d000 8c518000    vdrvroot     vdrvroot.sys
8c518000 8c529000    partmgr      partmgr.sys
8c529000 8c539000    volmgr       volmgr.sys
8c539000 8c584000    volmgrx      volmgrx.sys
8c584000 8c58b000    pciide       pciide.sys
8c58b000 8c599000    PCIINDEX     PCIINDEX.SYS
8c599000 8c5af000    mountmgr     mountmgr.sys
8c5af000 8c5d8180    vmbus        vmbus.sys
8c5d9000 8c5eb000    winhvc       winhvc.sys
8c5eb000 8c5f4000    atapi        atapi.sys
```

```
0: kd> lm n
start      end          module name
00d10000 00d72000    kd          kd.exe
5c930000 5ccb6000    dbgeng      dbgeng.dll
5ccc0000 5cde1000    dbghelp     dbghelp.dll
5d900000 5d948000    symsrv      symsrv.dll
74810000 74819000    VERSION     VERSION.dll
754b0000 754fa000    KERNELBASE  KERNELBASE.dll
75650000 75669000    sechost     sechost.dll
75700000 757a1000    RPCRT4       RPCRT4.dll
76bf0000 76cc4000    kernel32     kernel32.dll
76e70000 76f1c000    msvcrt       msvcrt.dll
76f20000 76fc0000    ADVAPI32     ADVAPI32.dll
771d0000 7730c000    ntdll        ntdll.dll
8ca00000 8ca11000    fileinfo     fileinfo.sys
8ca11000 8ca96000    mcupdate_GenuineIntel mcupdate_GenuineIntel
8ca96000 8caa7000    PSHEd        PSHEd.dll
8caa7000 8caaf000    BOOTVID      BOOTVID.dll
8caaf000 8caf1000    CLFS          CLFS.SYS
8caf1000 8cb9c000    CI            CI.dll
8cb9c000 8cbbf000    ataport       ataport.SYS
8cbbf000 8cbc8000    amdxtata      amdxtata.sys
8cbc8000 8cbfc000    fltmgr        fltmgr.sys
8cc00000 8cc09000    atapi         atapi.sys
8cc09000 8cc13000    msahci        msahci.sys
8cc19000 8cc8a000    Wdf01000      Wdf01000.sys
8cc8a000 8cc98000    WDFLDR        WDFLDR.SYS
8cc98000 8cce0000    ACPI          ACPI.sys
8cce0000 8cce9000    WMILIB        WMILIB.SYS
8cce9000 8ccf1000    msisadrv      msisadrv.sys
```



## 사용자 공간 구성 (User Space Topography)

### ▪ /BASE 링커 옵션

- EXE : 0x400000
- DLL : 0x10000000
- 메모리가 사용가능하지 않을 경우 재배치(relocation)
- /FIXED 옵션은 재배치 방지

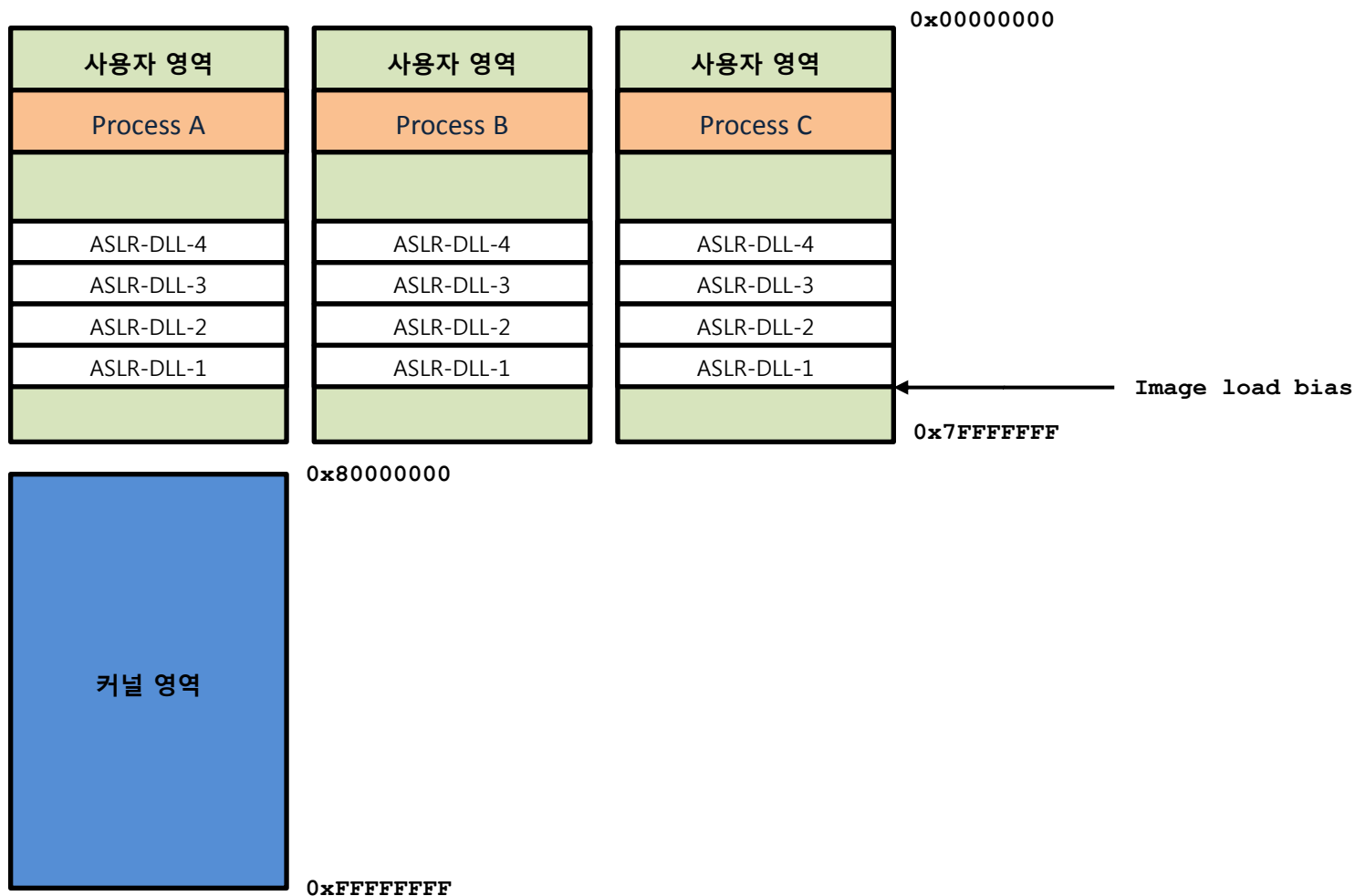
### ▪ /DYNAMICBASE (ASLR, Vista 이상 적용)

- 로드될 때 메모리 상의 임의 주소 배치
- MS 컴파일러가 아닐 경우, ASLR 적용을 위해서는 추가컴파일 필요



## 사용자 공간 구성 (User Space Topography)

### ▪ DLL 로드

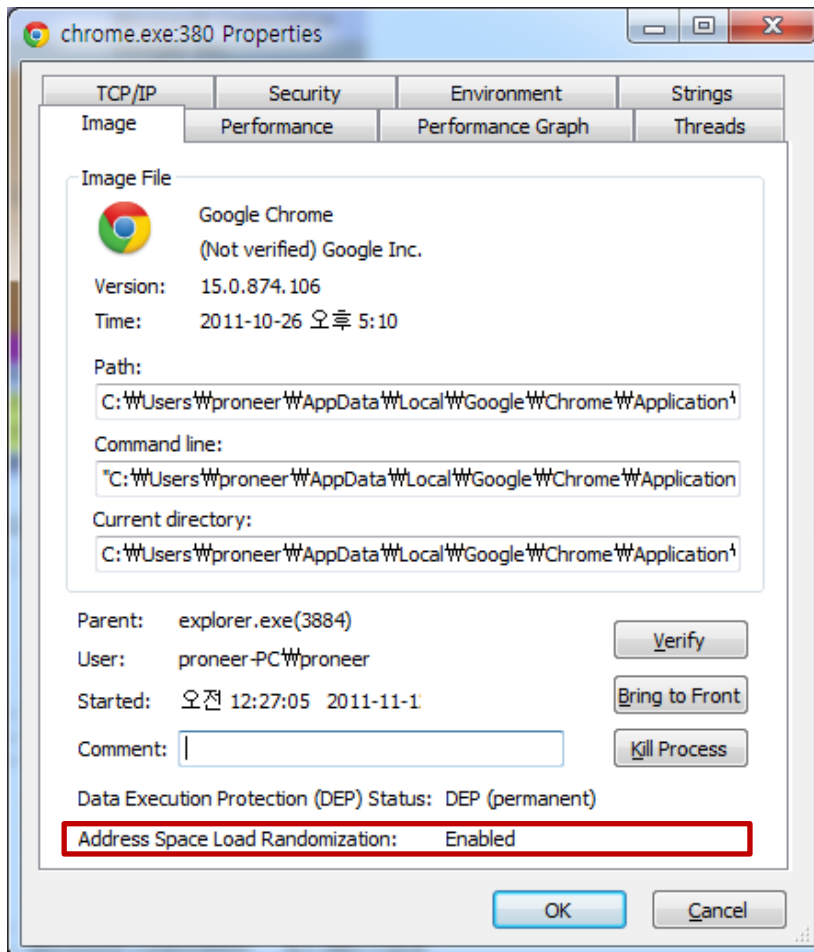






## 사용자 공간 구성 (User Space Topography)

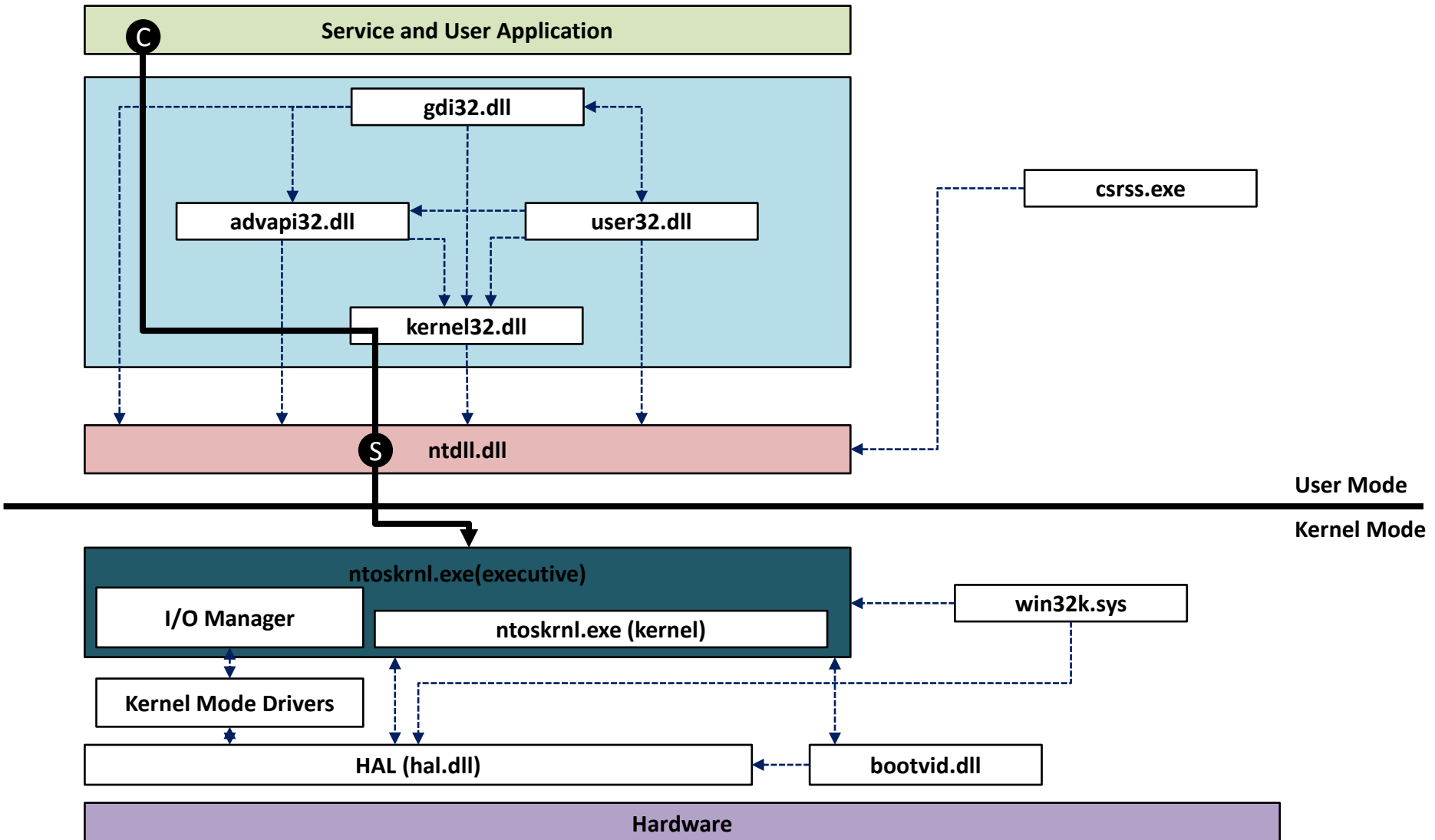
- Process Explorer



# User Mode vs. Kernel Mode

- How versus Where
- Kernel-Mode Components
- User-Mode Components

# 사용자 모드 vs 커널 모드





## 커널 모드 컴포넌트

### ■ 커널 관련 컴포넌트

| File Name    | Description                                                             |
|--------------|-------------------------------------------------------------------------|
| ntoskrnl.exe | Uniprocessor x86 architecture systems where <b>PAE</b> is not supported |
| ntkrnlpa.exe | Uniprocessor x86 architecture systems with <b>PAE</b> support           |
| ntkrnlmp.exe | Multiprocessor version of <b>ntoskrnl.exe</b>                           |
| ntkrpamp.exe | Multiprocessor version of <b>ntkrnlpa.exe</b>                           |

### ■ 커널 주요 컴포넌트 임포트 모듈 (dumpbin.exe 활용)

| Component    | Imported Modules                                             |
|--------------|--------------------------------------------------------------|
| hal.dll      | ntoskrnl.exe, kdcom.dll, pshed.dll                           |
| bootvid.dll  | ntoskrnl.exe, hal.dll                                        |
| ntoskrnl.exe | hal.dll, pshed.dll, bootvid.dll, kdcom.dll, clfs.sys, ci.dll |
| win32k.sys   | ntoskrnl.exe, msrpc.sys, watchdog.sys, hal.dll, dxapi.sys    |



## 사용자 모드 컴포넌트

### ▪ 서브시스템

- **Win32** (what Microsoft wanted to people to use)
- **WOW** (supported legacy Windows 3.1 apps)
- **NTVDM** (supported even older MS-DOS apps)
- **OS/2** (an attempt to appeal to the IBM crowd)
- **POSIX** (an attempt to appeal to the UNIX crowd)

### ▪ 서브시스템을 구성하는 주요 컴포넌트

- User-mode Client-Server Runtime Subsystem (csrss.exe)
- Kernel-mode device driver (win32k.sys)
- User-mode DLLs that implement the subsystem's API



## 사용자 모드 컴포넌트

- **User-mode Client-Server Runtime Subsystem (csrss.exe)**
  - 사용자 모드의 프로세스와 스레드를 관리하는 역할
  - CLI(Command Line Interface) 지원
  - 사용자 영역의 필수 구성 요소
- **사용자 애플리케이션에서 노출되는 서브시스템 인터페이스 (API)**
  - kernel32.dll, advapi32.dll, user32.dll, gdi.dll, shell32.dll, rpcrt4.dll, etc



## 사용자 모드 컴포넌트

- 사용자 주요 컴포넌트 임포트 모듈 (dumpbin.exe 활용)

| Component    | Imported Modules                                                                        |
|--------------|-----------------------------------------------------------------------------------------|
| advapi32.dll | ntdll.dll, kernel32.dll, user32.dll, rpcrt4.dll, wintrust.dll, secur32.dll, bcrypt.dll  |
| user32.dll   | ntdll.dll, kernel32.dll, gdi32.dll, advapi32.dll, msimg32.dll, powrprof.dll, winsta.dll |
| gdi32.dll    | ntdll.dll, kernel32.dll, user32.dll, advapi32.dll                                       |
| csrss.exe    | ntdll.dll, csrsrv.dll                                                                   |
| kernel32.dll | ntdll.dll                                                                               |
| ntdll.dll    | none                                                                                    |

