

chapter 7



c

가



“ ” 가 가?

“ 가? ”

```

“ ” ,“C 가? ”
” C
가 가 가? C
가
Hello, world

```

```

#include <stdio.h>
int main(void)
{
    printf(" Hello, world\n ");
    return 0;
}

```

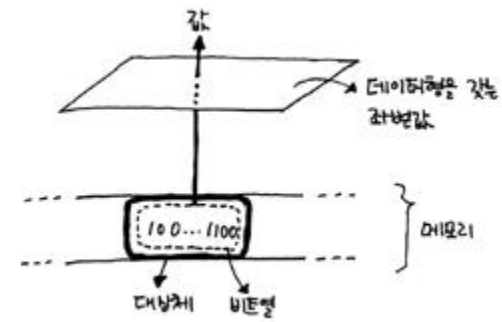
가 . 2

```

C
C C
가
*
가 C

```

‘ (value) ’ 가
 ()
 ()



가
 (object type)

(incomplete type)
 (function type)

void “ ”

가

가

‘ (value) ’

7-1

가 가

(rvalue)

가

가 ()

=

- sizeof : , sizeof 가
- & :
- ++ -- :
- :
- = :

()

()

가

가

```
[ 7-1]
01: int i = 8002;
02: const int j = 7903;
03:
04: int main(void)
05: {
06:   int k;
07:
08:   k = i;
09:   k = j;
10:
11:   return 0;
12: }
```

int i, k const int j가

8 9 가 , k =

, i, j

k

i j int const

int (8002 7903) . 8 ,

9

08: k = 8002;

09: k = 7903;

, const volatile

. int const

int ,

.8 i가 int .i가 int

가 가
u, l

C
309

int . ,j가
const int . , 가
7903
가 ?
가
int . ,j
int . (C ' 가
C ,) C ' 가
(modifiable lvalue) ' 가
const

[7-2] const 가

```
01: const int ci = 8402;
02:
03: int main(void)
04: {
05:   ci = 221; /* wrong */
06:
07:   return 0;
08: }
```

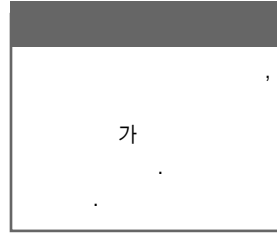
5 const
(ci 가) const
가
) 가
가 const

C
C
"lvalue"
가 C
C

[7-3]

```
01: int ai1[10];
02:
03: int main(void)
04: {
05:   int ai2[10];
06:
07:   ai2 = ai1; /* wrong */
08:   ai2[1] = ai1[1];
09:
10:   return 0;
11: }
```

ai1 ai2 array of int , (1 , 5) [] 10
C
8 ai1, ai2 [1]가 ai1, ai2
C
C
가 가
"lvalue" " " , C "lvalue "1"
()"locator"



const C 가

- const
- const

“ ” const

가

-
-

가 가

가

[7-3] array of

int ai2 ai2[1] (

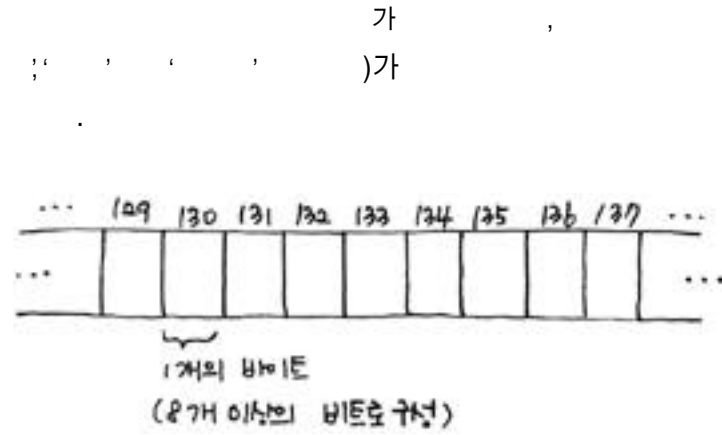
) int

가

C

가

CPU가 .C



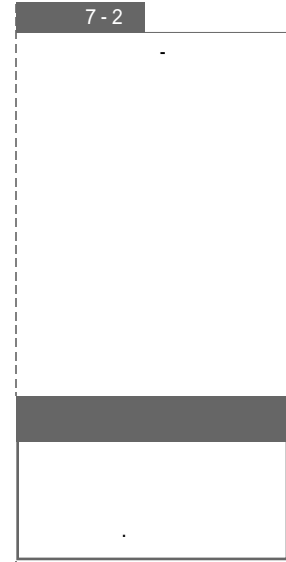
,가

가

(linear memory model)

가

가



가 . (object) (object

type)

C

가

가

“ ” “ ”

(가 가)“

가

2001 가 ,

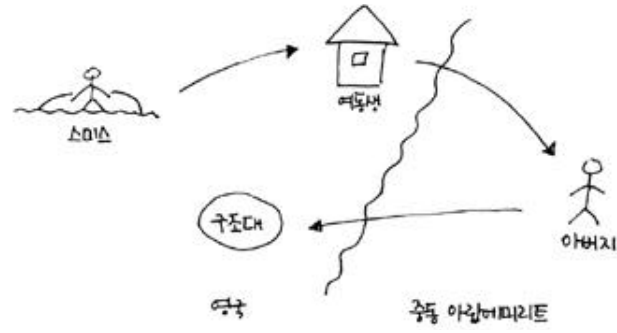
가

가

가

가

1



C
" "
가

)
!

가가

가

" "

()

가

(indirect memory access)
(indirect addressing)

이이이 ... 11011
정수형의 값은 어떤 주소를 의미하는 값으로 해석된다.

01110 ... 0011
포인터형의 값은 어떤 주소를 의미하는 값으로 해석된다.

가

()

()

가

).

()

가?

가

가

가?

가

가

{ pointer to }

가가

가

"...pointer"

"...pointer to ..."

가가

()

가

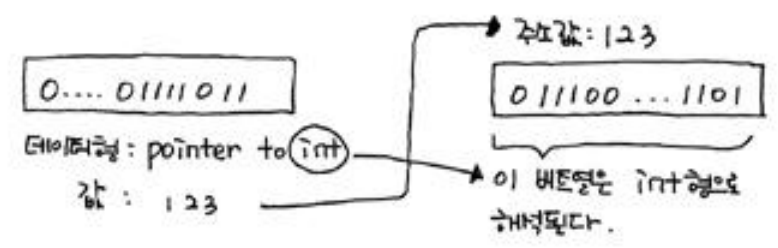
가

pointer to int(int

가

)

int



가 “ ” 가 .

(to to) 가 ?

- (call by reference) C (call by value)

- 가 가 (가)

- (subscript)
- (dynamic memory allocation) 가 가

- (tree) (linked list) (data structure) C



가 3 ,1 가 .

가 가?

가 , 가 C C가 (,)

ref 가 가 C 가 가

swap(ref a, ref b); /* a b */

, C99 가 VLA , C

가 , C 가 ? C 가

.C C

HOW?

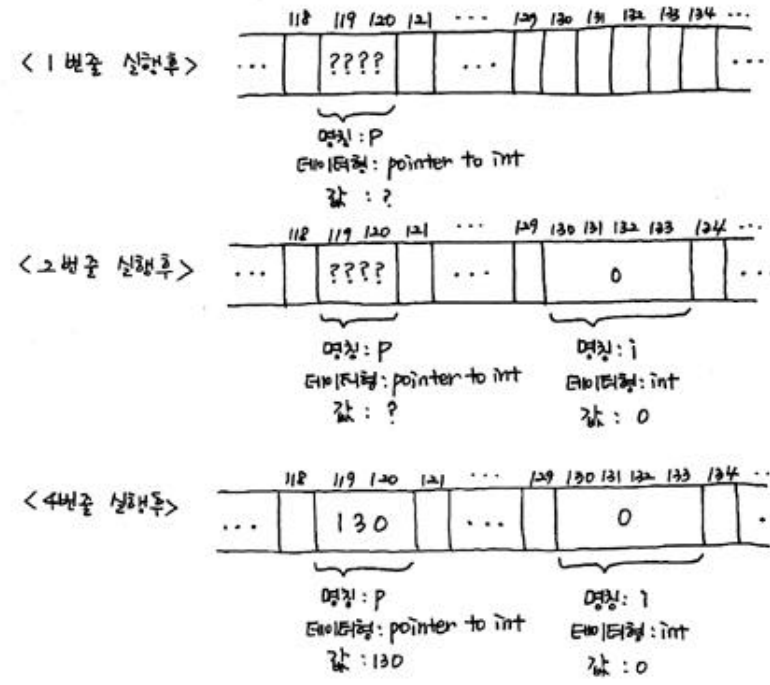




“address of”
“address of i”
가

```
01: int *p;
02: int i = 0;
03:
04: p = &i;
```

1, 2 p pointer to int(int) i
int . 4
pointer to int p 가(&i)
& (*)
& (address
operator) ,
, “pointer to” &i
i int
, “pointer to”int가 p
가? pointer to int .
“ ” 4 가 , p 가
i , “ p가 i 가 ”
, int 4 , pointer to int 2
가 .



, i int 2
0 . p 1 , 4
(&i) . p i

```
01: int i = 0;
02: int *p = &i;
```

가 가 . i p 가 ,
p . p i

```
01: int *p = &i; /* wrong */
02: int i = 0;
```

, 1 p가 &i(i) , 1
가 i , 1
i

*p &i 가? p &i
 ?
 01: int *p;
 02: int i = 0;
 03:
 04: *p = &i; /* wrong */
 4 가?
 .C 가 ,
 int *p = &i 가
 *p가 p . *p &i
 가
 * (ndirection operator) ' .
 (dereferencing)

	(&)	(*)
		가 가
	() "pointer to"	"pointer to"

HOW?



가 " " "indirection" "dereferencing"
가 .

indirection

가 .

[7-4] "indirection"

```

01: #include <stdio.h>
02:
03: int a[12] = { 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 };
04: int x[10] = { 9, 8, 7, 6, 5, 4, 3, 2, 1, 0 };
05:
06: int main(void)
07: {
08:     printf(" %d\n", a[x[2]]);
09:
10:     return 0;
11: }
    
```

8 , a[] (subscript)
x[] (a[])
(x[]) "indirection"

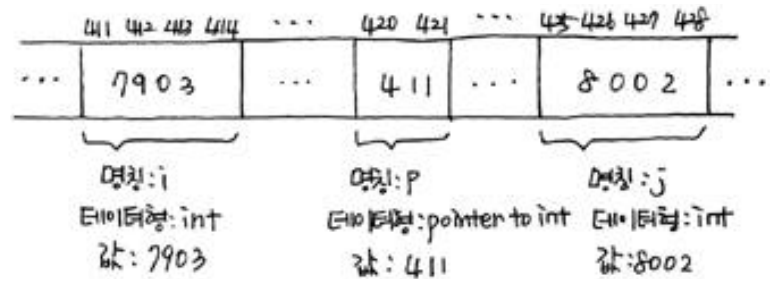
dereferencing

" " "indirection" "dereferencing"
 "dereferencing" C "indirection" 가
 "indirection" "dereferencing"
 " " C
 가 "dereferencing"
 , "indirection"

13 . *p 13

13: i = 7903;

*p가 p가 가 (i)
(가) i
13



14 가

*p &i
. *p 가
, *p int , &i pointer to int

.C



(alias) 가 (aliasing) 가
가 C 가 C99
restrict

[7-6]

```

01: #include <stdio.h>
02:
03: void swap(int a, int b)
04: {
05:     int t;
06:
07:     t = a;
08:     a = b;
09:     b = t;
10: }
11:
12: int main(void)
13: {
14:     int i = 8002;
15:     int j = 7903;
16:
17:     printf(" Before: %d and %d\n ", i, j);
18:     swap(i, j); /* not swapped */
19:     printf(" After: %d and %d\n ", i, j);
20:
21:     return 0;
22: }

```

i j swap()

17 가

Before: 8002 and 7903

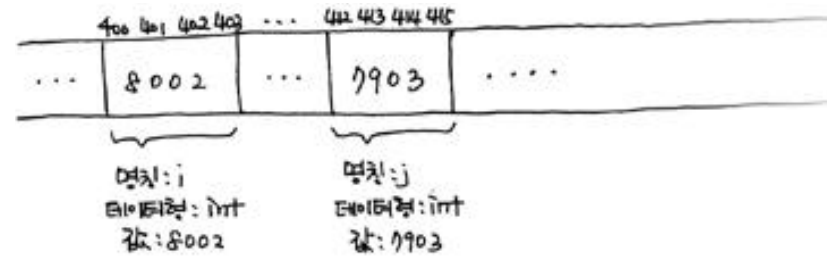
19

After: 7903 and 8002

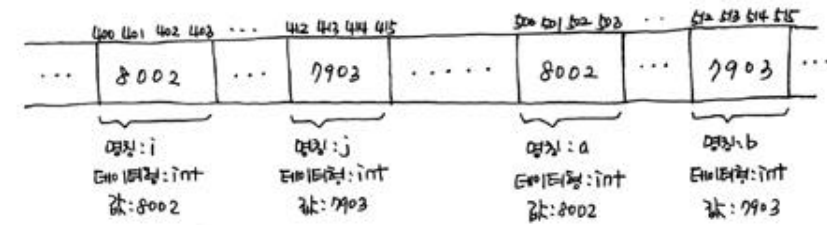
C 가 ()
 i, j a, b가)
 ' (call by reference) '

C 가 가 ,
 ' (call by value) ' swap()

main() i, j가 () i j 가
 int 2 가 .



18 " " swap() , swap()
 () a, b 가 , i, j
 a, b .



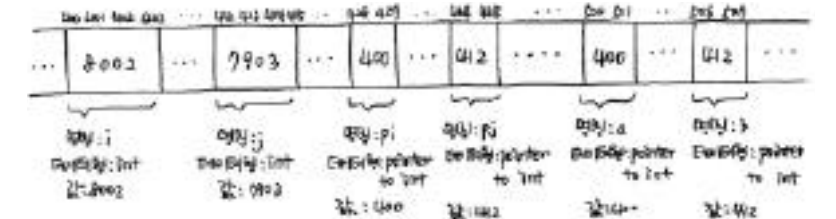
[7-6] swap() t a
 b . swap() a, b i, j
 . swap() a, b ,
 i, j , swap() a, b
 a, b 가 .

main() swap() swap()
 swap() 가 .

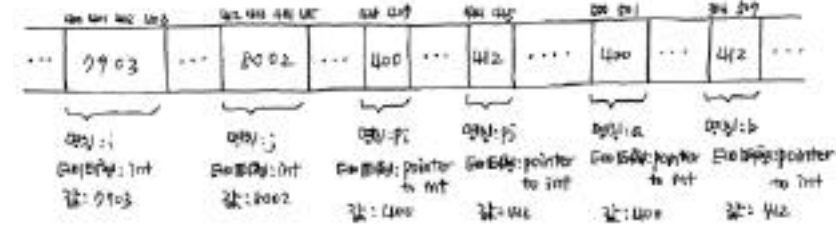
```
[ 7-7]
01: #include <stdio.h>
02:
03: void swap(int *a, int *b)
04: {
05:     int t;
06:
07:     t = *a;
08:     *a = *b;
09:     *b = t;
10: }
11:
12: int main(void)
13: {
14:     int i = 8002;
15:     int j = 7903;
16:     int *pi = &i;
17:     int *pj = &j;
18:
19:     printf(" Before: %d and %d\n ", i, j);
20:     swap(pi, pj);
21:     printf(" After: %d and %d\n ", i, j);
22:
23:     return 0;
24: }
```

가 가 . swap() 가int
 pointer to int , swap() a, b가

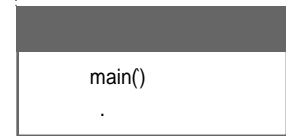
swap() ,
 i, j pi, pj
 가
 swap() 20 , 가 i j
 pi, pj
 3 swap() a, b pointer to int
 .C
 [7-6]
 가
 swap()



swap() pi pj i, j
 가 i, j가
 가 7, 8, 9
 i, j swap()



C
 가가
 “ ”
 가
 [7-7] i j swap()
 pi pj
 가 i, j
 swap() .18
 12: int main(void)
 13: {
 14: int i = 8002;
 15: int j = 7903;
 16:
 17: printf("Before: %d and %d\n", i, j);
 18: swap(&i, &j);
 19: printf("After: %d and %d\n", i, j);
 20:
 21: return 0;
 22: }
 [7-7] 가 pi, pj swap()
 ,C
 pi, pj (i, j)
 가



[7-6] swap() a, b i, j
 가 swap()
 가 i, j , main() i, j
 가
 “ ”
 가

가 가 가

[7-8]

```

01: int main(void)
02: {
03:   int *p;
04:
05:   {
06:     int i;
07:
08:     p = &i;
09:   }
10:   printf("%p\n", (void *)p); /* wrong */
11:
12:   return 0;
13: }

```

8 pointer to int p 6 i 가 .
 9 } i 가
 . i 가 p
 i 가
 . 가가 가
 . 가가 (,10
)
 . 가가 가
 가

10
 , 10
 .
 10: p; /* wrong */
 ' ' .
 " p " 10 ,
 " " ? 가
 ,가 .
 " " . 가가
 ?

Q: , 가가 () ?

```
printf("%d\n", *p); /* i */
```

A: () () () 가)
 , 가가 () 가



가
 (가) 가 가
 가



C가

가

(type conversion) ' .int
 pointer to int
 pointer to int int

(가) “ ”

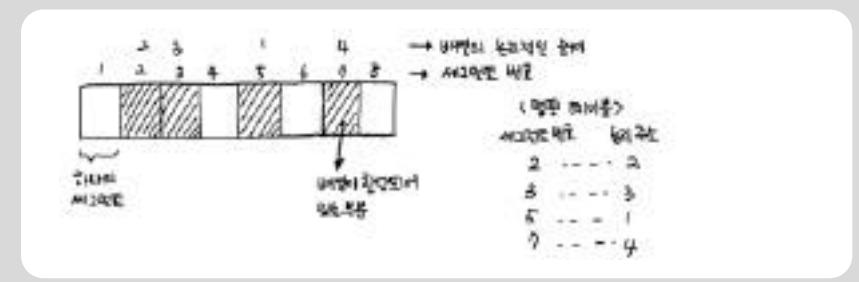
,4
 8002 , 3.14159가

“type punning”

가

“punning (同音異義)

가 (segmented architecture)
 (segment descriptor) ' (offset)
 가 가
 가



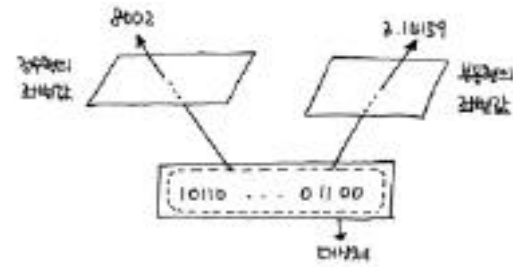
[7-6]

, C
 가
 2 가5 C
 가

HOW?



7-7
type punning



가

[7-9]

```

01: int main(void)
02: {
03:   double d = 3.0;
04:   int i;
05:
06:   i = (int) d;
07:
08:   printf(" %d\n ", i);
09:   printf(" %fn ", d);
10:
11:   return 0;
12: }

```



6 double 3.0 int i
가
()가
6 (int) 가 double 3.0
int (cast
operator)

int 3 가
double 3.0 int
, double 3.0 int 3
가 int i
" "
• 가 .C
가
" "
•
d
3.0
int 3
. 9
, ,C
,6 가
(int) 가
(int)

가 ? 가
 가가 ? 가
 980
 980
 C가
 (?)
 가)
 가
 가 가
 가 가 long int unsigned long int
 unsigned long int
 unsigned long int
 가

C
 , C 가
 가 B BCPL . B BCPL
 " " , B BCPL " "
 , B BCPL 가
 . B BCPL
 가 . C B, BCPL
 , B, BCPL

가
 가

```
[ 7-10]
01: int main(void)
02: {
03:   int i;
04:   int j = 7903;
05:   int *p = &j;
06:
07:   i = (int) p; /* wrong - */
08:   p = (int *) j; /* wrong - */
09:
10:   return 0;
11: }
```

01 C99

가
 , volatile
 가?
 가 가?
 int 가
 0x80000000 가
 int *p = (int *) 0x80000000;
 가 가

p 가 가 ,
 가 가
 가 가
 , 가
 , ()
 가
 C
 C 가 , , ,
 C가 , , ,
 ,
 가 가C99
 int int ,int const int
 , 가
 ?pointer to int , pointer to
 double ?
 가 ,
 가 .

가 (가)“ pointer to ”
 ,
 ,가
 . pointer to int pointer to double int double
 . 가 pointer to int
 pointer to const int . [5-2]
 , pointer to char, pointer to signed char, pointer to unsigned char
 . pointer to signedshort pointer to short int
 (signed short short int가)
 ,pointer to pointer to int pointer to function returning int
 (pointer to int function returning int가)
 가
 가 . , pointer to function returning int
 pointer to function(pointer to char, int) returning int 가
 . function returning int
 function(pointer to char, int) returning int가
 가
 C 가 “ ” 가
 . 가 가
 가
 가
 , int const int가 , pointer to char
 const pointer to char
 , 가
 가
 , 가

uniform
 pointer ' .
 2 pointer to
 (signed/unsigned) char 4 가
 Prime
 pointer to char가 가 ,HP 3000 64-bit
 Cray
 가
 가

C
 가가 (, ,
)
 C가
 ()
 가
 가
 (alignment requirement)



HOW?



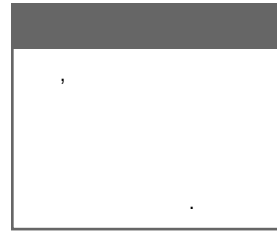
pointer to int const pointer to int가
 int const int가
 (,)
 ()
 pointer to int, pointer to const int, pointer to volatile int,
 pointer to const volatile int int const int int

가?

C 가 가 ,
 C
 가 32 가 가
 8
 가 8 가
 가 4
 pointer to (unsigned/signed) char
 4 가
 32 pointer to
 char 가 , 32
 pointer to char (가)
 pointer to char가 (가)

HOW?





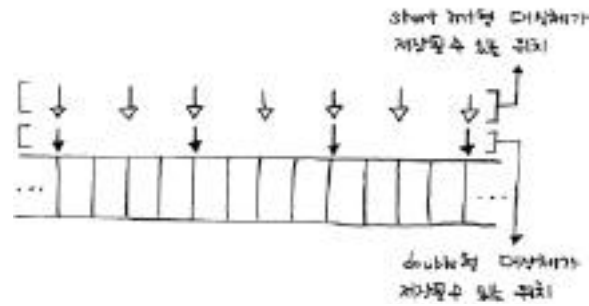
```

01: short int si, *psi;
02: double d, *pd;
03:
04: psi = (short int *) &d; /* wrong */
05: pd = (double *) &si; /* wrong */

```

short int 2, double 8 가 ,
 2 4 가 .
 “ ” ,
 “ ” 가 short int
 2 가 , double 4
 가 . pointer to short int , pointer to double
 2 , 4 가 .
 short int 가 , double 가

7-8



4 . &d double d
 , pointer to double . 4
 가 . (d) 16 가 . pointer to double
 16 (short int *) pointer to short int

pointer to short int 2
 16 2 , pointer
 to short int 4 ,
 가 .
 5 가 &si
 short int si , pointer to short int .
 2 가 , 6 가
 . pointer to short int 6 (double *)
 pointer to double . pointer to double
 4 가 6 4 가 ,
 pointer to double 가 .
 ,
 가 .
 , 6 pointer to double
 . double 4
 , 6 (pointer to double)
 double , 가
 가 .
 가 . , pointer to double 6
 4 8
 . 6
 ,
 . pointer to double pointer
 to short int , 6
 , 4 8
 , 4 2 pointer to double
 pointer to short int .
 2 4 가 가 가
 . 가

double 1 ()
 가
 4 ,5
 가 .1 가
 1
 pointer to (unsigned/signed) char 1 가
 pointer to (unsigned/signed) char

```
01: int i;
02: double d;
03: unsigned short int usi;
04: char *pc;
05:
06: pc = (char *) &i;
07: pc = (char *) &d;
08: pc = (char *) &usi;
```

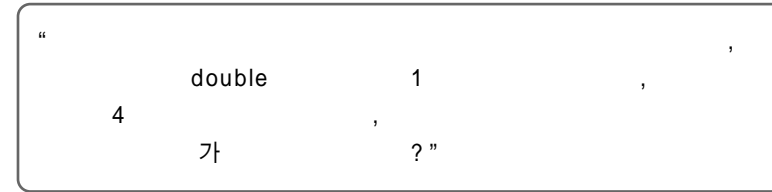
, pointer to (unsigned/signed) char

```
01: int i, *pi;
02: char *pc;
03:
04: pc = (char *) &i;
05: pi = (int *) pc;
```

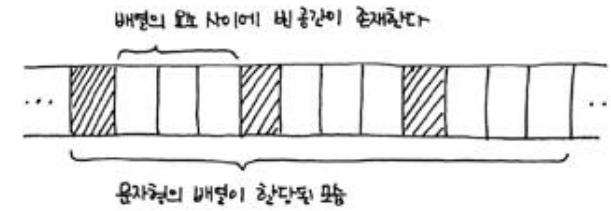
4 pointer to char pointer to int (&i) 5
 pointer to int pi . pi
 &i . 4 ,5
 (pc)

```
05: pi = (int *)((char *) &i);
```

((char *) &i) pointer to char " "
 (int *) pointer to int



가
 C
 가,C
 1
 4 가



3
 가
 1

```

01: int i;
02: const int *pci;
03: unsigned int *pui;
04:
05: pci = (const int *) &i;
06: pui = (unsigned int *) &i;

```

가
, C 가

가
“type punning”

가 , “type punning”



HOW?



pointer to array of T pointer to T

가 () 가
, 가

```

01: int arr[10];
02: int *p;
03:
04: p = (int *) &arr;

```

, &arr 가 , pointer to array of int가
. array of int int
. array of int(int)
int . int
int , int
int , int
가 가 int 가
int 가 int 가
array of int int
. pointer to array of int pointer to int

[7-11-1] “type punning”

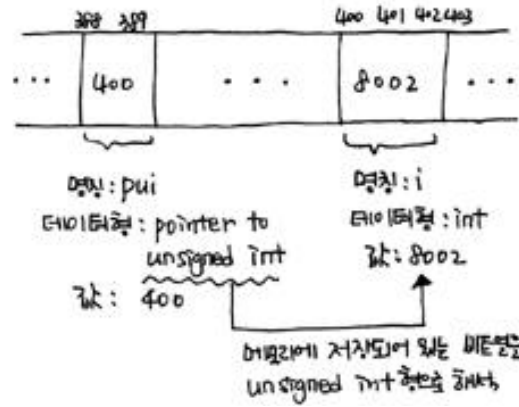
```

01: #include <stdio.h>
02:
03: int main(void)
04: {
05:     int i;
06:     unsigned int *pui;
07:
08:     i = 8002;
09:     pui = (unsigned int *) &i;
10:
11:     printf(“ %u\n ”, *pui);
12:
13:     return 0;
14: }

```

9 pointer to int pointer to
unsigned int pui

pui 가 , 가 i
int unsigned int



11
unsigned int ,
? 가가
? ,

8002가
,9 11 pui
11: printf("%u\n", *((unsigned int *)&i);

((unsigned int *)&i) pointer to unsigned int
i (&i) , " "
i unsigned int

Q: int unsigned int
?

A: , int 가 (INT_MAX) unsigned int 가
(UINT_MAX) int
unsigned int unsigned int

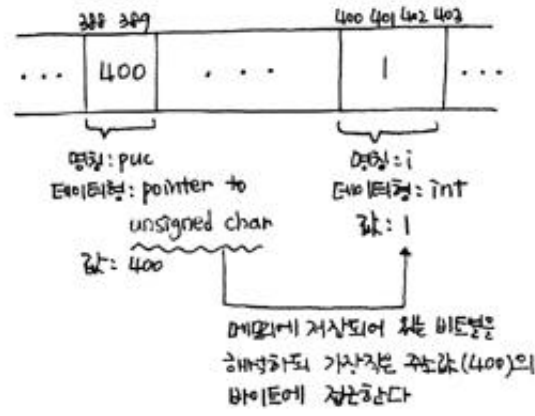
"type punning"

[7-11-2] "type punning"

```
01: #include <stdio.h>
02:
03: int main(void)
04: {
05:     int i = 1;
06:     unsigned char *puc;
07:
08:     puc = (unsigned char *) &i;
09:     printf( "%d\n", (int) *puc);
10:
11:     return 0;
12: }
```

		(int)		
(unsigned int)	,		가	
(unsigned char)			8	
pointer to int	pointer to unsigned char		puc	
.	int	가4	,	1
가	,	puc int		4
			가	
가		가		





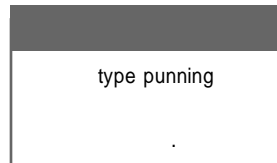
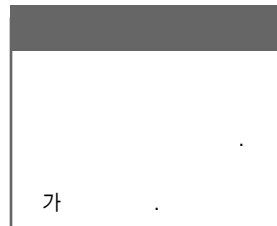
9 puc , i
가 unsigned char

- 가 ?
- (endian)
- ()

type punning (unsigned char)

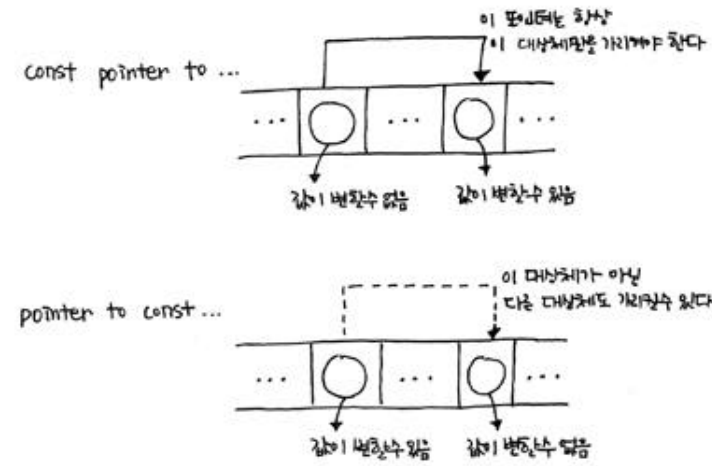
(aliasing)

type punning



(more strict) " 가 n , m double , 2 int " short

가 const pointer to ... pointer to const ... , const pointer to ... 가 가 가 , pointer to const ... 가 가 가



가 const pointer to ... , const int int , const pointer to ...

7-9 const pointer to ... pointer to const ...

pointer to const
 (가가)
 . 가
 .
 가가 가
 가 . pointer to const ... const ... 가
 . , const int 가
 pointer to const int .
 const int i;
 const int *pci = &i;
 , pci 가 const int
 pci = 8402; / wrong */
 int i;
 const int *pci = &i;
 가 . i
 . pointer to int .
 pointer to int pci pointer to const
 int . 가 ,
 , const int int
 .
 (const int *) ,
 . 가 가
 . const
 가 가? " 가
 " 가? . const
 가 , const
 가 . const (const int *)
 가 const (int) 가 가 . C

“ ” 가
 . ,C
 가
 strcpy()
 char *strcpy(char *s1, const char *s2);
 pointer to char pointer to const
 char . 가가
 가 , 가가
 .
 가 가 , 가 가
 ,C 가
 ,
 strcpy() (const char *)
 . const const
 가 .
 const int i;
 int *pci = &i; /* wrong */
 const “ 가 “
 ” . pointer to const int pointer to
 int , 가 . const
 가 const 가
 . 가가
 . C
 ,
 .
 const int i;
 int *pci = (int *) &i;

```
(
  "returning"
) char *
```