

# Topic V1B

Parameter Passing by Value and by Reference

Reading: (Section 2.14)

What is the  
Difference Between  
Passing a Parameter  
by Value and Passing  
it by Reference?

# The car.c Program

```
#include <stdio.h>
#define STRINGLENGTH 20

typedef struct c_node{
    int    vehicleID;
    char   make[STRINGLENGTH];
    char   model[STRINGLENGTH];
    int    year;
    int    mileage;
    double cost;
    struct c_node *next;
} CarNode;

void ReadCar(CarNode *car);
void PrintCar(CarNode car);

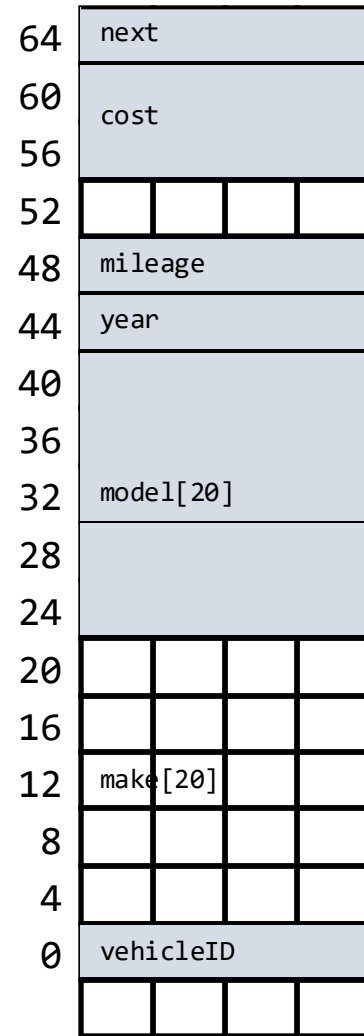
main(){
    CarNode mycar;
    ReadCar(&mycar);
    PrintCar(mycar);
}
```

```
void ReadCar(CarNode *car){
    car->vehicleID = 2;
    strcpy(car->make, "Chrysler");
    strcpy(car->model, "Town and Country");
    car->year = 2014;
    car->mileage = 6000;
    car->cost = 45,625.74;
}

void PrintCar(CarNode car){
    printf("vehicleID:%d\n", car.vehicleID);
    printf("make:%s\n", car.make);
    printf("model:%s\n", car.model);
    printf("year:%d\n", car.year);
    printf("mileage:%d\n", car.mileage);
    printf("cost:%f\n", car.cost);
}
```

```
#define STRINGLENGTH 20
```

```
typedef struct c_node{  
    int    vehicleID;  
    char   make[STRINGLENGTH];  
    char   model[STRINGLENGTH];  
    int    year;  
    int    mileage;  
    double cost;  
    struct c_node *next;  
} CarNode;
```



```

#include <stdio.h>
#define STRINGLENGTH 20

typedef struct c_node{
    int    vehicleID;
    char   make[STRINGLENGTH];
    char   model[STRINGLENGTH];
    int    year;
    int    mileage;
    double cost;
    struct c_node *next;
} CarNode;

void ReadCar(CarNode *car);
void PrintCar(CarNode car);

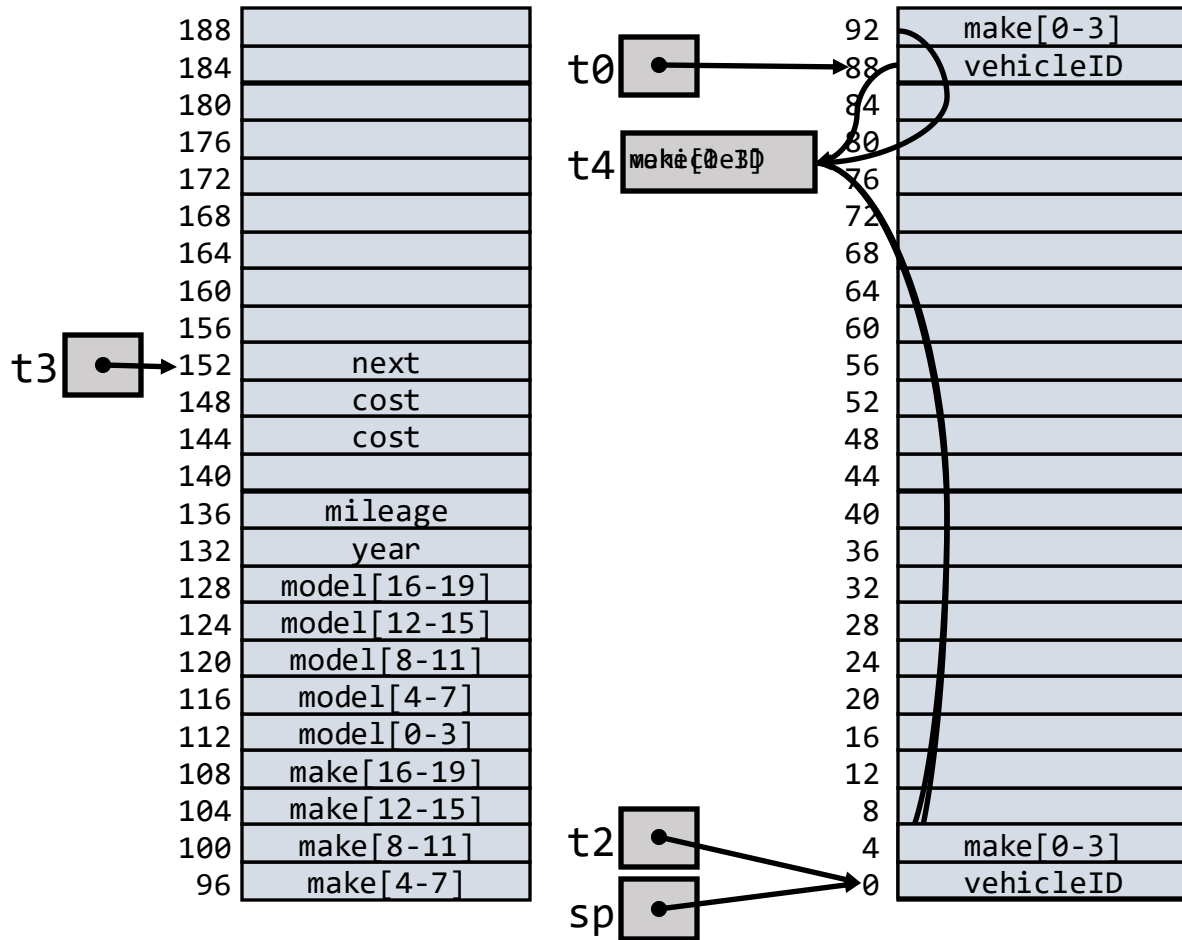
main(){
    CarNode mycar;
    ReadCar(&mycar);
    PrintCar(mycar);
}

```

```

main:
    addi    sp, -168
    sw      ra, 84(sp)
# 23 CarNode mycar;
# 25 ReadCar(&mycar);
    add     a0, sp, 88
    jal     ra, ReadCar
# 27 PrintCar(mycar);
    add     t0, sp, 88
    mv      t2, sp
    add     t3, t0, 72
L1:        lw      t4, 0(t0)
    sw      t4, 0(t2)
    addi    t0, t0, 4
    addi    t2, t2, 4
    bne     t0, t3, L1
    jal     ra,
PrintCar
# 28
    }
    mv      a0, zero
    lw      ra, 84(sp)
    addi    sp, 168
    jalr    zero, ra, 0

```



```

main:
    addi    sp, -168
    sw      ra, 84(sp)
# 23 CarNode mycar;
# 25 ReadCar(&mycar);
    add     a0, sp, 88
    jal     ra, ReadCar
# 27 PrintCar(mycar);
    add     t0, sp, 88
    mv      t2, sp
    add     t3, t0, 72
L1:
    lw      t4, 0(t0)
    sw      t4, 0(t2)
    addi    t0, t0, 4
    addi    t2, t2, 4
    bne     t0, t3, L1
    jal     ra,
PrintCar
# 28
    }
    mv      a0, zero
    lw      ra, 84(sp)
    addi    sp, 168
    jalr    zero, ra, 0

```

188	
184	
180	
176	
172	
168	
164	
160	
156	
152	next
148	cost
144	cost
140	
136	mileage
132	year
128	model[16-19]
124	model[12-15]
120	model[8-11]
116	model[4-7]
112	model[0-3]
108	make[16-19]
104	make[12-15]
100	make[8-11]
96	make[4-7]

92	make[0-3]
88	vehicleID
84	
80	
76	
72	
68	
64	next
60	cost
56	cost
52	
48	mileage
44	year
40	model[16-19]
36	model[12-15]
32	model[8-11]
28	model[4-7]
24	model[0-3]
20	make[16-19]
16	make[12-15]
12	make[8-11]
8	make[4-7]
4	make[0-3]
0	vehicleID

```

main:
    addi    sp, -168
    sw      ra, 84(sp)
    # 23 CarNode mycar;
    # 25 ReadCar(&mycar);
        add    a0, sp, 88
        jal    ra, ReadCar
    # 27 PrintCar(mycar);
        add    t0, sp, 88
        mv     t2, sp
        add    t3, t0, 72
L1:     lw     t4, 0(t0)
        sw     t4, 0(t2)
        addi   t0, t0, 4
        addi   t2, t2, 4
        bne    t0, t3, L1
        jal    ra,
PrintCar
# 28    }
        mv     a0, zero
        lw     ra, 84(sp)
        addi   sp, 168
        jalr   zero, ra, 0

```



# Why so Much Copying?

The program `car.c` passes the data structure `CarNode` to the `PrintCar` function **by value**

A copy of each byte of `CarNode` must be made in the stack for each call of the function `PrintCar`

Instead, we could have passed the address of the copy of `CarNode` that we already had in the stack

```
#include <stdio.h>
#define STRINGLENGTH 20

typedef struct c_node{
    int    vehicleID;
    char   make[STRINGLENGTH];
    char   model[STRINGLENGTH];
    int    year;
    int    mileage;
    double cost;
    struct c_node *next;
} CarNode;

void ReadCar(CarNode *car);
void PrintCar(CarNode *car);

main(){
    CarNode mycar;
    ReadCar(&mycar);
    PrintCar(&mycar);
}
```

```
void ReadCar(CarNode *car){
    car->vehicleID = 2;
    strcpy(car->make, "Chrysler");
    strcpy(car->model, "Town and Country");
    car->year = 2014;
    car->mileage = 6000;
    car->cost = 45,625.74;
}

void PrintCar(CarNode *car){
    printf("vehicleID:%d\n", car->vehicleID);
    printf("make:%s\n", car->make);
    printf("model:%s\n", car->model);
    printf("year:%d\n", car->year);
    printf("mileage:%d\n", car->mileage);
    printf("cost:%f\n", car->cost);
}
```

```

#include <stdio.h>
#define STRINGLENGTH 20

typedef struct c_node{
    int    vehicleID;
    char   make[STRINGLENGTH];
    char   model[STRINGLENGTH];
    int    year;
    int    mileage;
    double cost;
    struct c_node *next;
} CarNode;

void ReadCar(CarNode *car);
void PrintCar(CarNode *car);

main(){
    CarNode mycar;
    ReadCar(&mycar);
    PrintCar(&mycar);
}

```

```

main:
        addi    sp, -112
        sw      ra, 28(sp)
# 23 CarNode mycar;
# 25 ReadCar(&mycar);
        add     a0, sp, 32
        jal     ra, ReadCar
# 27 PrintCar(&mycar);
        add     a0, sp, 32
        jal     ra,
PrintCar
# 28      }
        mv      a0, zero
        lw      ra, 28(sp)
        addi    sp, 112
        jalr    zero, ra, 0

```