Topic V1B

Parameter Passing by Value and by Reference

Reading: (Section 2.14)

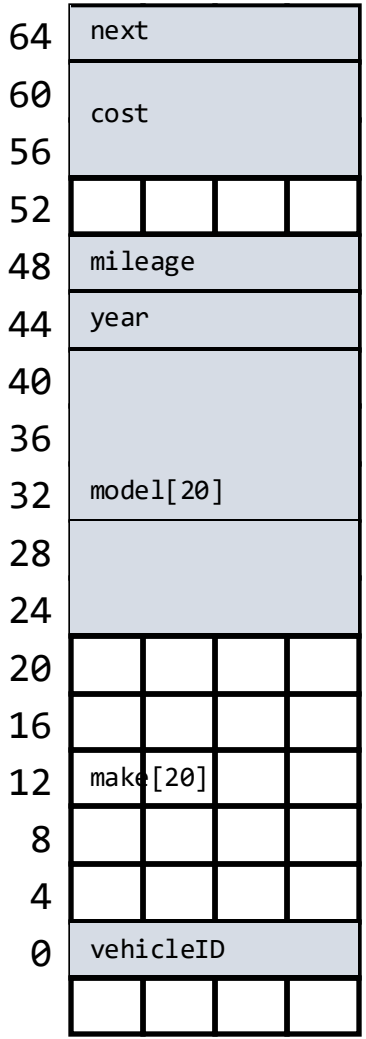
What is the Difference Between Passing a Parameter by Value and Passing it by Reference?

The car.c Program

```c
#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);
}
```

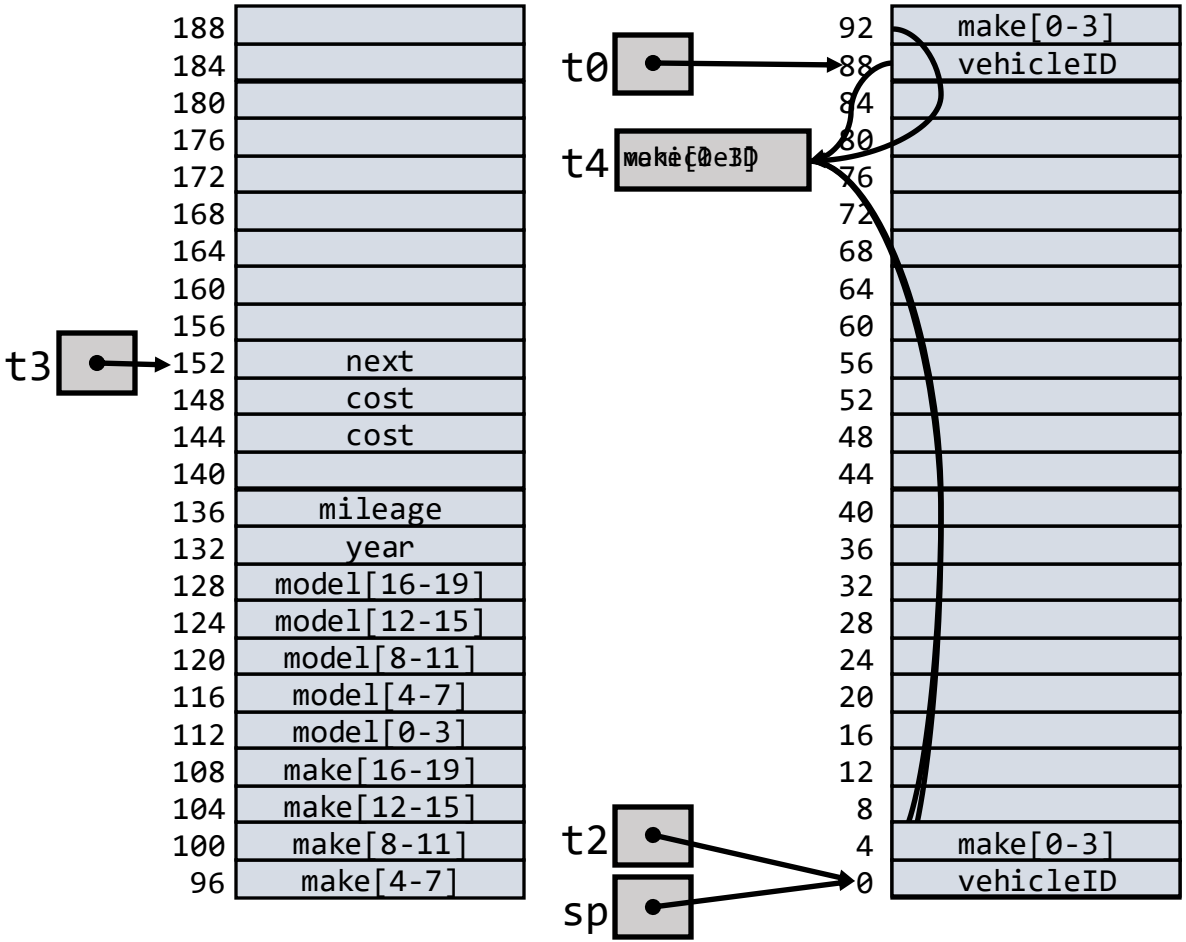
```c
void ReadCar(CarNode *car){
  car->vehicleID = 2;
  strcpy(car->make, “Chrysler”);
  strcpy(car->model, “Town and Country”);
  car->year = 2014;
  car->mileage = 6000;
  car->cost = 45,625.74;
}

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);
}
```

```c
#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;
```

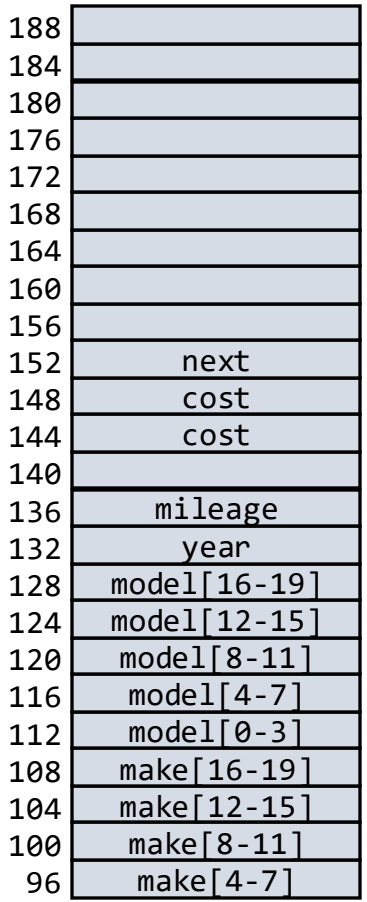


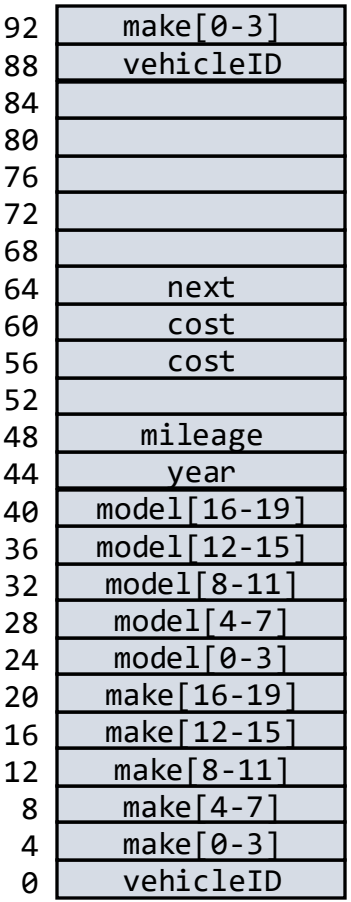
```c
#include <stdio.h>
#define STRINGLENGTH 20

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
```






```
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

```c
#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);
}
```

```c
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);
}
```

```c
#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
```