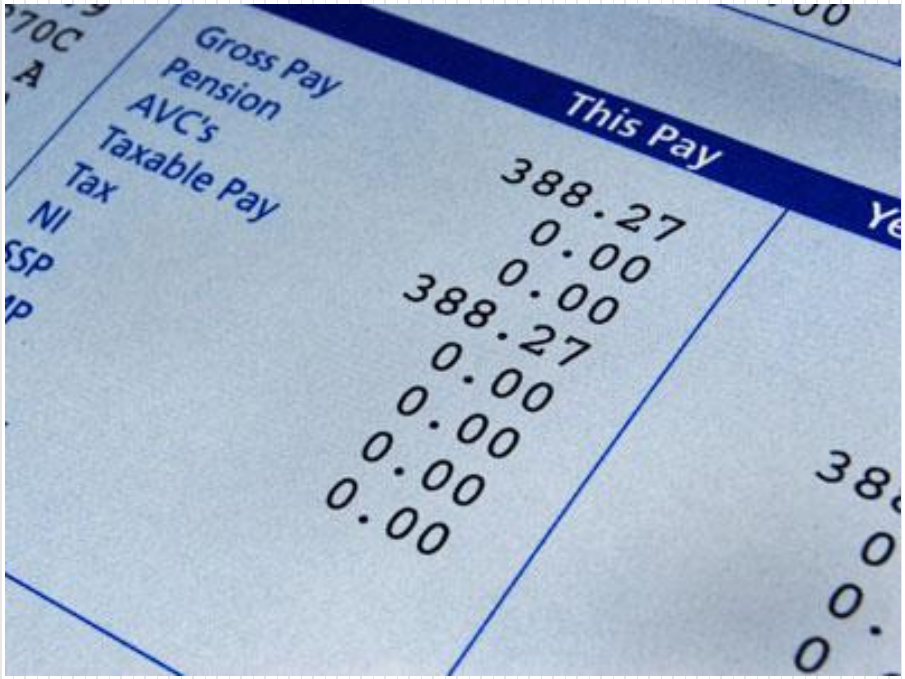
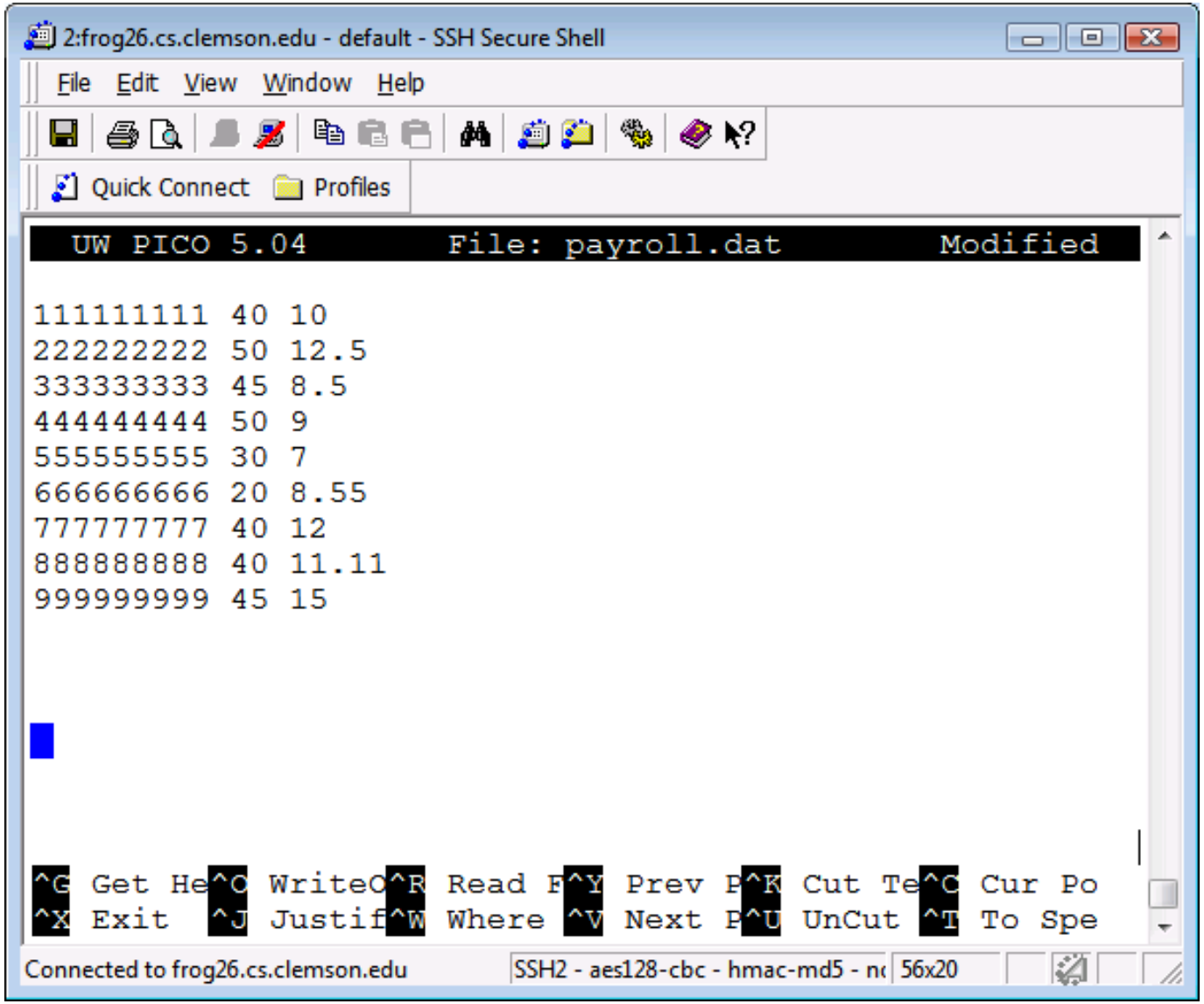
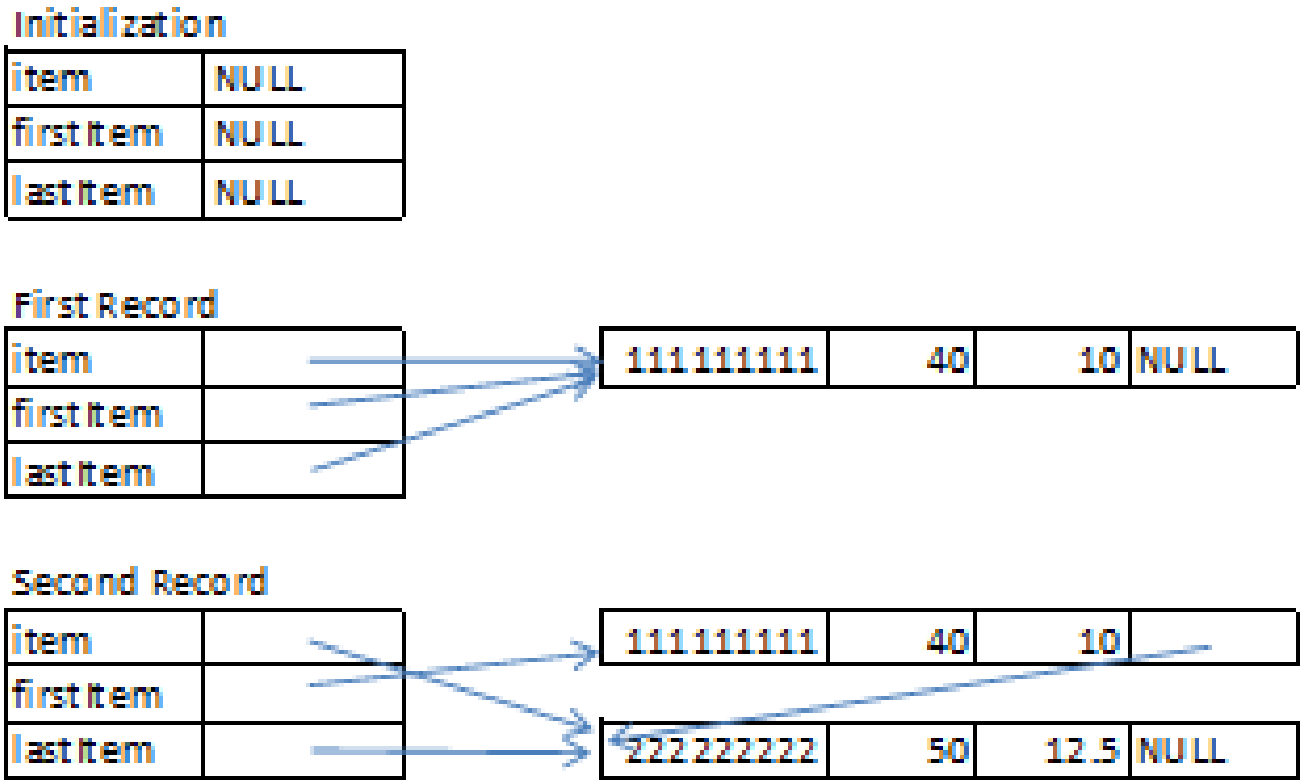
Programming in C

Chapter 8 App
Payroll Lookup using Structure

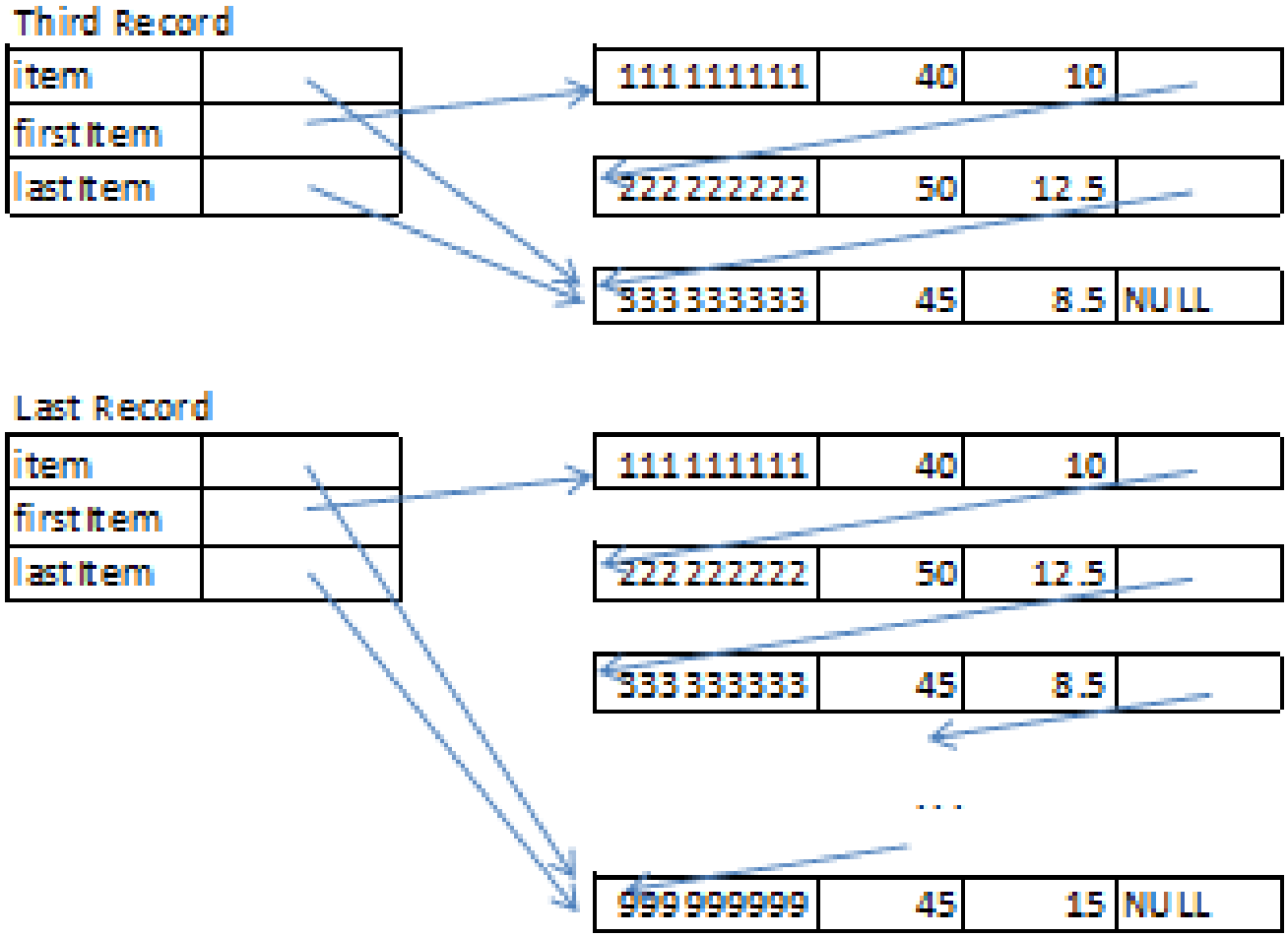



2:frog26.cs.clemson.edu - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
UW PICO 5.04 File: payroll.dat Modified
111111111 40 10
222222222 50 12.5
333333333 45 8.5
444444444 50 9
555555555 30 7
666666666 20 8.55
777777777 40 12
888888888 40 11.11
999999999 45 15
^G Get He ^O WriteO ^R Read F ^Y Prev P ^K Cut Te ^C Cur Po
^X Exit ^J Justif ^W Where ^V Next P ^U UnCut ^T To Spe
Connected to frog26.cs.clemson.edu
SSH2 - aes128-cbc - hmac-md5 - n
56x20

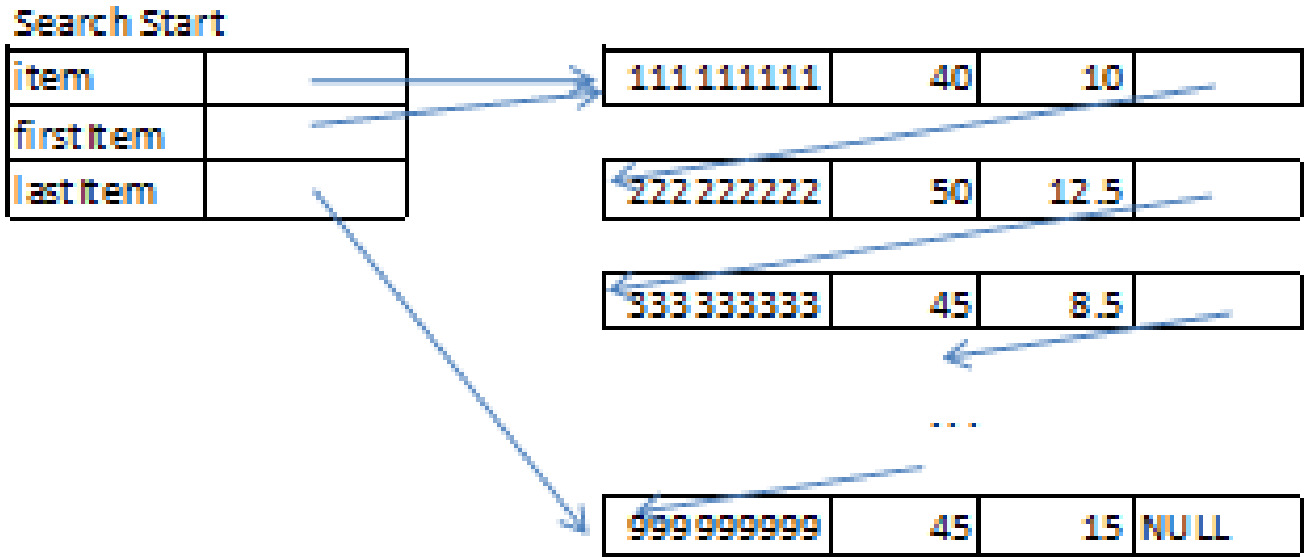
Memory Allocation & Structure



Memory Allocation & Structure



Traversing Records



ch08PayType.h

```
/*
Header:    ch08PayType
Purpose:   Definition of pay_t
Author:    Ima Programmer
Date:      mm/dd/yy
*/

struct pay_t {
   long id;
   int hrs;
   float rate;
   struct pay_t * nextPtr;
};
```

ch08Pay.c

```
/*
Program:  ch08Pay
Purpose:  Lookup payroll record using struct
Author:   Ima Programmer
Date:     mm/dd/yy
*/

#include <stdio.h>
#include <stdlib.h>
#include "ch08PayType.h"

int main(int argc, char *argv[])
{
   // initialization
   struct pay_t * item = NULL;
   struct pay_t * firstItem = NULL;
   struct pay_t * lastItem = NULL;

   FILE * payFile = NULL;
   long id;
   float hrs, rate, pay;
```

```
// Open input
if (argc != 2) {
   printf("\nInvalid number of arguments\n\n");
   exit(1);
}
payFile = fopen(argv[1], "r");
if (payFile == NULL) {
   printf("\nCannot open %s\n\n", argv[1]);
   exit(2);
}
```

```
// load payroll data
while (fscanf(payFile, "%ld %f %f", &id, &hrs, &rate) == 3) {
   // get new pay item
   item = malloc(sizeof(struct pay_t));
   if (item == NULL) {
      printf("\nUnable to allocate memory for %ld!\n\n", id);
      exit(3);
   }
   if (firstItem == NULL)
      firstItem = item;
   else
      (*lastItem).nextPtr = item;
            // or lastItem->nextPtr = item
   lastItem = item;

   // load item data
   item->id = id;              // or (*item).id = id
   item->hrs = hrs;
   item->rate = rate;
   item->nextPtr = NULL;
} // load
fclose(payFile);
```

```
// lookup: process ids until zero
printf("\nEnter id or zero to end: ");
scanf("%ld", &id);
while (id != 0) {
   // lookup item
   item = firstItem;
   while (item != NULL && item->id != id)
      item = item->nextPtr;

   if (item == NULL) // not found
       printf("%ld not found\n", id);
   else { // found
       if ((*item).hrs <= 40)
          pay = item->hrs * item->rate;
       else
          pay = 40 * item->rate + (item->hrs - 40) * item->rate * 1.5;
       printf("Hours = %d, Rate = %f, Pay = %.2f\n",
             item->hrs, item->rate, pay);
   }

   // next id
   printf("\nEnter id or zero to end: ");
   scanf("%ld", &id);
}
```

```
    // free memory
    while (firstItem != NULL) {
        item = firstItem;
        firstItem = item->nextPtr;
        free(item);
    }

    printf("\n");
    return 0;   // normal return
} // main
```

Programming in C

Chapter 8 App
Payroll Lookup using Struct

THE END