

CSI 333 – Programming at the Hardware-Software Interface

Example on Linked Lists

Handout 5.3

Program Example 9: The following C program creates a linked list. Each node of the list contains an integer value and a pointer to the next node of the list.

The program reads positive integer values typed in by a user one at a time, and inserts a node containing that value at the end of the linked list. When the user inputs zero or a negative value, the program stops accepting input, prints the values in the linked list and terminates.

```
#include <stdio.h>
#include <stdlib.h>

/* Struct for each node in the linked list. */
struct node {
    int value;
    struct node *next;
};

int main(void) {

    /* Prototypes for the two functions used. */
    void print_list(struct node*);
    void insert_node(struct node**, struct node**, int);

    /* Pointers to the first and last nodes of list are used */
    /* to facilitate insertion at the end of list.          */
    struct node *head, *tail;

    int i; /* To read input value */
    head = tail = NULL; /* List is initially empty. */

    printf("Next value = "); /* Prompt the user for input. */
    scanf("%d", &i);

    while (i > 0) {
        /* Input value is positive. Insert a new node with value */
        /* of i at the end of the list. The insert_node function */
        /* may modify the values of head and tail.                */
        insert_node(&head, &tail, i);
        printf("Next value = ");
        scanf("%d", &i); /* Read next input value */
    }
    /* End of input. Print the list. */
    print_list(head);
    return 0;
} /* End of main. */
```

```

void insert_node (struct node **h, struct node **t, int v) {

    /* Creates a new node with value given by parameter v */
    /* and inserts that node at the end of the list whose */
    /* first and last nodes are pointed to by *h and *t */
    /* respectively. */

    struct node *temp;

    if ((temp = (struct node *)malloc(sizeof(struct node))) == NULL) {
        printf("Node allocation failed. \n");
        exit(1); /* Stop program */
    }
    /* Space for node obtained. Copy the value v into the node */
    /* and insert the node at the end of the list. */

    temp->value = v; temp->next = NULL;
    if (*h == NULL) {
        /* List is currently empty. */
        *h = *t = temp;
    }
    else { /* The list is not empty. Use *t to add the node */
        /* at the end. */
        (*t)->next = temp; *t = (*t)->next;
    }
} /* End of insert_node. */

void print_list(struct node *h) {

    /* Prints the values stored in the nodes of the list */
    /* pointed to by h. */

    if (h == NULL) {
        printf("The list is empty.\n");
    }
    else {
        printf("Values in the list are:\n");
        while (h != NULL) {
            printf("%d\n", h->value);
            h = h->next;
        }
    }
} /* End of print_list */

```