CPNA Lecture 23 - Variable Length Arguments

Mridul Sankar Barik

Jadavpur University

2023

Introduction

- It is often desirable to implement a function where the number of arguments is not known, or is not constant, when the function is written
- <`stdarg.h`> library provides us with the `va_list` data type, as well as the macros `va_start`, `va_arg`, and `va_end` for manipulating the list of arguments

va_list Data Type

- va_list is a data type defined in <stdarg.h> to hold the list of arguments passed into the function

```
va_list args;
```

Macros

- va_start is a macro used to initialize the argument list so that we can begin reading arguments from it

```
va_start(args, num);
```

- va_arg is the macro used to read an argument from the list; two parameters
  - the va_list object we created, args
  - a data type va_arg will return the next argument as this type
- va_end is another macro that cleans up our args object for us when we're done using it

Points to Remember

- There must be at least one named argument before the ellipsis
- You can specify as many named arguments as you'd like as long the ellipsis comes after all of them
- If specifying more than one named argument, the argument name passed as the second parameter to `va_start()` must be the **last** (right-most) named argument

Example I

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

int max(int num, ...){
    int i, temp, m;
    va_list args;

    va_start(args, num);

    m=va_arg(args, int);
    for(i=2;i<=num; i++){
        temp=va_arg(args, int);
        if(m<temp)
            m=temp;
    }
    return(m);
}

main(){
    printf("Max = %d\n", max(4, 23, 56, 89, 29));
}
```

Example II

```
void myprintf(char *fmtstr, ...){
    va_list args;
    va_start(args, fmtstr);

    while(*fmtstr!=’\0’){
        while(*fmtstr!=’%’){
            if(*fmtstr!=’\0’)
                return;
            printf("%c", *fmtstr);
            fmtstr++;
        }
        fmtstr++;
        switch(*fmtstr){
            case ’d’: printf("%d", va_arg(args, int));break;
            case ’f’: printf("%f", va_arg(args, double));break;
            case ’c’: printf("%c", va_arg(args, int));break;
            case ’s’: printf("%s", va_arg(args, char *)); break;
        }
        fmtstr++;
    }
}
main(){
    myprintf("Hello %s %d", "World", 2019);
}
```