

CSCI 2132
Software Development

Lecture 25:
String Library Functions

Instructor: Vlado Keselj

Faculty of Computer Science

Dalhousie University

Previous Lecture

- Finished mergesort2 “fill-in the blanks”
- Strings:
- String literals
- String variables
- Reading and writing strings

Aside: A Common Mistake

```
/* Example vla.c */
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    /* double a[n];      /* Wrong! */
    while (n > 0) {
        double a[n];    /* Correct! */
        printf("Size of double: %d\n", sizeof(double));
        printf("Size of array: %d\n", sizeof(a));
        printf("Length of array: %d\n",
               sizeof(a) / sizeof(double));
        /* Array can be used here */
        scanf("%d", &n);
    }
    return 0; }

```

More Examples

- Reading next 19 characters (any characters including whitespace):

```
char s[20];  
scanf("%19c", s);  
s[19] = '\\0';
```

- Read up to 19 letters (whitespace not ignored, null character appended):

```
char s[20];  
scanf("%19[a-zA-Z]", s);
```

Reading a Line

- Function `gets` can be used to read input line, but not recommended due to possible buffer overflow
- Better to use `fgets`; example:

```
char line[1000];  
fgets(line, 1000, stdin);
```

Buffer Overflow Risks

- Happens if we read input without length control, for example:

```
char buffer[1000];  
gets(buffer);  
/* or */  
scanf("%s", buffer);
```

- Bug, but may also be a risk of security attack
- A malicious user can provide a long, carefully crafted input to overwrite stack and cause program to start executing injected machine code

Example of a User-Implemented Read Function

```
/* Read one line from stdin and store in string str of
   capacity n.
   The null character is included in the capacity n.
   If the line is too long, only a part is read.
*/
int read_line(char str[], int n) {
    int ch, i = 0;

    while ( (ch = getchar()) != '\n' && ch != EOF )
        if (i < n-1) str[i++] = ch;

    str[i] = '\0';

    return i;
}
```

String Library Functions

- Required include line:

```
#include <string.h>
```

- String copy:

```
char* strcpy(char *s1, const char *s2);
```

- **Warning:** Make sure to have sufficient memory; example of **wrong** code:

```
char s1[] = "abc";  
strcpy(s1, "defg");
```

- String concatenation

```
char* strcat(char *s1, const char *s2);
```

- **Warning:** Sufficient memory required

String Library Functions (continued)

- String comparison:

```
int strcmp(const char *s1, const char *s2);
```

- String length

```
size_t strlen(const char *s1);
```