C-STRINGS AND NUMBER CONVERSION

Documentation and Examples

Delroy A. Brinkerhoff

C-STRINGS AND THE CONSOLE

SYSTEM CALLS

- `read(char* buf, int number);`
  - `read(void* buf, size_t number);`
- `write(char* buf, int number);`
  - `write(void* buf, size_t number);`

CONVERSION OPERATORS

- >>
  - Calls the read system call
  - Converts C-strings to numbers
- <<
  - Converts numbers to C-strings
  - Calls write system call

C-STRING (ASCII) TO NUMBER CONVERSIONS

DOCUMENTATION PROTOTYPES

- int atoi(const char* str);
- long atol(const char* str);
- double atof(const char* str);

EXAMPLE FUNCTION CALLS

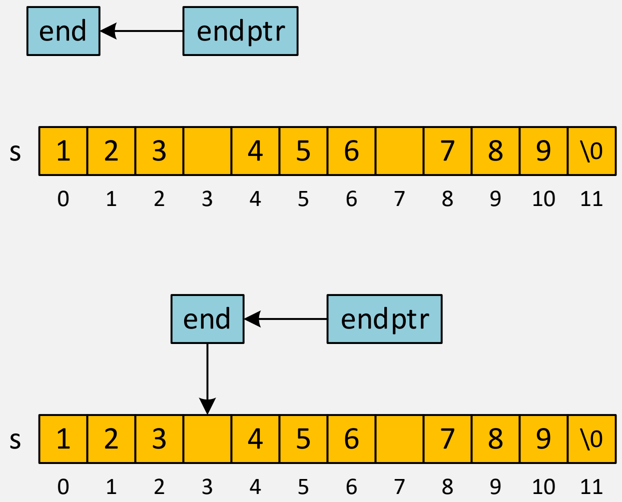
- cout << atoi(s1) << endl;
- cout << atoi("123") << endl;
- cout << atol(s1) << endl;
- cout << atol("123") << endl;
- cout << atof(s2) << endl;
- cout << atof("3.14159") << endl;

FLEXIBLE (ADVANCED) C-STRING TO NUMBER CONVERSIONS

- `long strtol(const char* index, char** endptr, int base);`
- `cout << strtol("123", nullptr, 10) << endl;`
- `cout << strtol("0xafcd", nullptr, 16) << endl;`

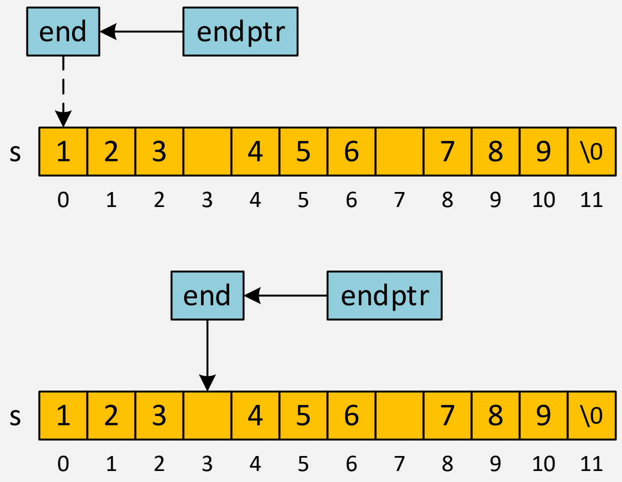
- `double strtod(const char* index, char** endptr);`
- `cout << strtod("3.14159", nullptr) << endl;`

THE endptr (I)



```
char s[] = "123 456 789";
char* end = nullptr;
cout << strtol(s, &end, 10) << endl;
cout << strtol(end, &end, 10) << endl;
cout << strtol(end, &end, 10) << endl;
```

THE endptr (2)



```
char s[] = "123 456 789";
char* end = s;
while (*end != '\0')
    cout << strtol(end, &end, 10) << endl;
```

NUMBERS TO C-STRINGS

- `char s[25];`
- `char* itoa(int num, char* str, int base);`
- `char* _itoa(int num, char* str, int base);`
- `errno_t _itoa_s(int num, char* str, size_t size, int base);`
- `itoa(123, s, 10);`
- `itoa(0xaf48, s, 16);`
- `_itoa_s(123, s, 25, 10);`
- `_itoa_s(0xaf48, s, 25, 16);`