PP 2005 © Walery Susłow

Podstawy programowania komputerów

Wykład 12: „Łańcuchy znakowe”

swalover@ie.tu.koszalin.pl

Łańcuchy znakowe: wprowadzenie

- Łańcuch znakowy to ciąg składający się z jednego lub więcej znaków.
- Język C nie posiada specjalnego typu łańcuchowego.
- Łańcuchy są przechowywane w tablicach zbudowanych z elementów typu char. Tablica ta musi być o jedną komórkę większa niż ilość znaków w łańcuchu. Ostatni element tablicy /0 to znak ASCII o kodzie równym 0, służy on do oznaczenia końca łańcucha.
- Efektywna pojemność łańcuchów zakończonych zerem (ang. null-terminated string), ograniczona jest wielkością dostępnej pamięci, w praktyce to 64kB.

Stała łańcuchowa

Przykładowe dane: `„Jak pies na łańcuchu.”`

- ✓ Znaki cudzysłowu nie są częścią łańcucha.
- ✓ Aby w ramach łańcucha użyć cudzysłowu należy stosować kombinację `\”`, np. `„\”Uciekaj!\”\n”`.
- ✓ Stała łańcuchowa należy do klasy statycznej – przechowywana jest w pamięci przez cały czas działania programu i tylko w jednej kopii, niezależnie od ilości razy wywoływania funkcji.

Inicjalizacja łańcucha

Skrócona i pełna forma standardowej składni inicjalizacji tablicy:

```
char S1 [ ] = „Ala ma kota.”;
char S2 [ ] = {‘A’,‘l’,‘a’,‘ ’,‘m’,‘a’,
               ‘ ’,‘k’,‘o’,‘t’,‘a’,‘.’,‘\0’};
```

W deklaracji łańcucha można użyć zapisu wskaźnikowego:

```
char *S3 = „Ala ma kota.”;
```

Ilość pamięci przeznaczona na łańcuch jest ustalana przez kompilator i nie zależy od sposobu inicjalizacji. Przy wykorzystaniu wskaźnika dozwolona jest inkrementacja, np. ma sens instrukcja `++S3;`

Wczytywanie łańcuchów

Funkcja gets()

- pobiera znaki ze standardowego urządzenia wejścia aż do napotkania \n - znaku nowej linii, generowanego przy wciśnięciu klawisza ENTER:

```
char imie[MAX];
printf(„Jak masz na imię?\n");
gets(imie);
```

- dodaje na końcu łańcucha znak zerowy (\0);
- nie sprawdza czy dane wejściowe zmieszczą się w zarezerwowanym obszarze (znaki nadmiarowe umieszcza w kolejnych komórkach pamięci).

Wczytywanie łańcuchów, cd.

Funkcja fgets()

- pozwala określić maksymalną liczbę znaków, przeznaczona jest przede wszystkim do obsługi wejścia/wyjścia plikowego;
- pobiera drugi argument określający maksymalną liczbę znaków do odczytania;
- gdy odczyta znak nowej linii to pozostawia go w łańcuchu w przeciwieństwie do gets(), która go odrzuca;
- pobiera trzeci argument - nazwę pliku, z którego mają zostać pobrane dane. Aby wczytać łańcuch z klawiatury, należy użyć argumentu stdin.

Wczytywanie łańcuchów, cd.

Funkcja scanf()

- można je określić jako funkcję pobierającą jedno słowo;
- w przypadku użycia specyfikatora `%s` łańcuch obejmuje wszystkie znaki aż do kolejnego znaku nie drukowanego (bez tego znaku);
- w przypadku określenia szerokości pola (`%10s`) funkcja `scanf()` kończy odczytywanie w momencie napotkania znaku niedrukowalnego lub w momencie pobrania podanej liczby znaków.

Wyświetlanie łańcuchów

Język C zawiera trzy standartowe funkcje biblioteczne służące do wyświetlania łańcuchów: puts(), fputs() i printf().

- Aby skorzystać z funkcji `puts()`, wystarczy jej przekazać adres łańcucha jako argument, funkcja zatrzymuje się w momencie napotkania znaku zerowego.
- Funkcja `fputs()` pobiera drugi argument określający plik, do którego należy zapisać dane, aby wyświetlić dane na ekranie należy użyć argumentu stdout (funkcja nie dodaje do danych wyjściowych znaku nowej linii).

Funkcje łańcuchowe

Biblioteka języka C udostępnia kilka funkcji przetwarzających łańcuchy. Ich prototypy znajdują się w pliku nagłówkowym *string.h*

Najbardziej użyteczne funkcje to: strlen(), strcat(), strncat(), strcmp(), strncmp(), strcpy(), strncpy() oraz sprintf() z pliku nagłówkowego `stdio.h`

Funkcje łańcuchowe, cd.

Funkcja *strcat()* i *strncat()*

- Funkcje „string concatenation” pobierają jako argumenty dwa łańcuchy.
- Kopia drugiego łańcucha zostaje przyłączona na końcu pierwszego; całość staje się nowym pierwszym łańcuchem, drugi łańcuch nie ulega zmianie.
- Funkcja *strcat()* należy do typu char * - wskaźnik do znaku.
- Ona nie sprawdza, czy drugi łańcuch zmieści się w pierwszej tablicy.
- Funkcja *strncat()* pobiera drugi argument, określając maksymalną liczbę znaków jakie mogą być dodane.

Funkcje łańcuchowe, cd.

Funkcje *strcmp()* i *strncmp()*:

- służą do porównywania łańcuchów;
- strncmp() porównuje znaki łańcuchów do momentu wykrycia różnicy lub do momentu porównania ilości znaków określonej przez trzeci argument:
- `if (strncmp(lista[i], „Kowal", 5) == 0`
- `licznik++;`

- Funkcje strcpy() i strncpy() są odpowiednikiem operatora przypisania dla zmiennych łańcuchowych.

Badanie długości łańcuchów

```
#include < stdio.h >
#include < string.h >      /* dostarcza prototypu dla strlen() */
#define POCHWALA „Masz wspaniałe imię!„

int main(void)
{ char imie[40];
  printf("Jak masz na imie?\n");
  scanf("%s", imie);
  printf("Witaj, %s. %s\n", imie, POCHWALA);
  printf("Twoje %d-literowe imię zajmuje %d komórek pamięci.\n",
                        strlen(imie), sizeof imie);
  return 0;
 }
```

Działania na łańcuchach

Inne techniki operowania na łańcuchach przy użyciu:

- stałej łańcuchowej,
- tablicy typy char,
- wskaźnika do char,
- tablicy łańcuchów znakowych.

```
char napis[] = „Dowolny napis";
char *p = "Jakiś tam napis";
//łańcuch tekstowy o zerowej długości
char *z = "";
```

Działania na łańcuchach, cd.

- Bardziej wymyślne operacje na łańcuchach są to przeszukiwanie, wyciąganie podłańcuchów (substring), sortowanie, konwersja.
- Wymagają one użycia specjalnie do tego celu przeznaczonych funkcji, np. przy zamianie liter na duże (capitals lub upper case):
- `strupr() // String to upper case,`
- lub przy wyszukiwaniu wyrazu na liście:
- `strstr(Lista, „kot");// Szukamy kota☺`