Prelucrări iterative

Marius Minea

28 martie 2006

Scrierea ciclurilor

În conceperea programelor care conţin cicluri:
– identificăm ce variabilă se modifică în fiecare iteraţie
– identificăm care e condiţia de continuare / oprire
– nu uităm instrucţiunea care modifică acea variabilă (altfel ciclul se va continua la infinit)

La ieşirea dintr-un ciclu, condiţia e falsă $\Rightarrow$ aceasta ne spune ceva despre valorile posibile ale variabilelor din condiţie.
Folosim această informaţie pentru a gândi mai departe programul.

Verificăm programul executând mental întâi un exemplu simplu
şi apoi la rulare, *teste* pentru diferitele situaţii posibile

Ciclul cu test final

`do` *instrucţiune* `while ( expresie );`

– uneori ştim sigur că un ciclu trebuie executat cel puţin o dată
– ca şi ciclul cu test iniţial, execută *instrucţiunea* atât timp cât valoarea expresiei e nenulă (adevărată).
– expresia se evaluează însă *după* fiecare iteraţie
– echivalent cu:

instrucţiune
`while (` *expresie* `)` *instrucţiune*

Instrucţiunea `break`

– produce ieşirea din corpul instrucţiunii `while`, `do`, `for` sau `switch` *imediat înconjurătoare*; execuţia continuă cu instrucţiunea următoare
– mai convenabilă decât testarea unei variabile boolene la ciclul următor
– mai lizibil, dacă codul peste care se sare e complex

```
int c, wcnt = 0;
do {  // elimina spatiile initiale, cel mult pana la '\n'
  do c = getchar(); while (isspace(c) && c != '\n');
  if (c == '\n' || c == EOF) break; // gata linia
  wcnt = wcnt + 1; // incepe un nou cuvant
  // citeste caractere noi cat timp nu sunt spatii albe sau EOF
  do c = getchar(); while (!isspace(c) && c != EOF);
  // daca am ajuns la sfarsit de linie sau EOF iesim
} while (c != '\n' && c != EOF);
```

Exemplu: găsirea de factori primi

```
#include <stdio.h>
unsigned primfact(unsigned n, unsigned d) { // cel mai mic factor prim
  while (n % d != 0) {     // incepand de la d
    d = d + 1;
    if (d > n/2) return n;  // nu mai pot fi alti factori pana la n
  }
  return d;     // cand iesim din while, sigur n % d == 0
}
void printfact(unsigned n) {
  unsigned d = 2;     // primul numar prim
  while (n > 1) {
    d = primfact(n, d);     // salvam valoarea pentru primul factor gasit
    printf("%u ", d);     // si data viitoare continuam de acolo
    n = n / d;
  }
  putchar('\n');
}
int main(void) {
  printfact(108);
  return 0;
}
```