Hallgatói Tudásbázis

http://hat.sch.bme.hu
Általában a parancssori argumentumokról

A parancssori argumentumok C-beli szintaktikája a következő

```
int main(int argc, char **argv)
{
…
}
```

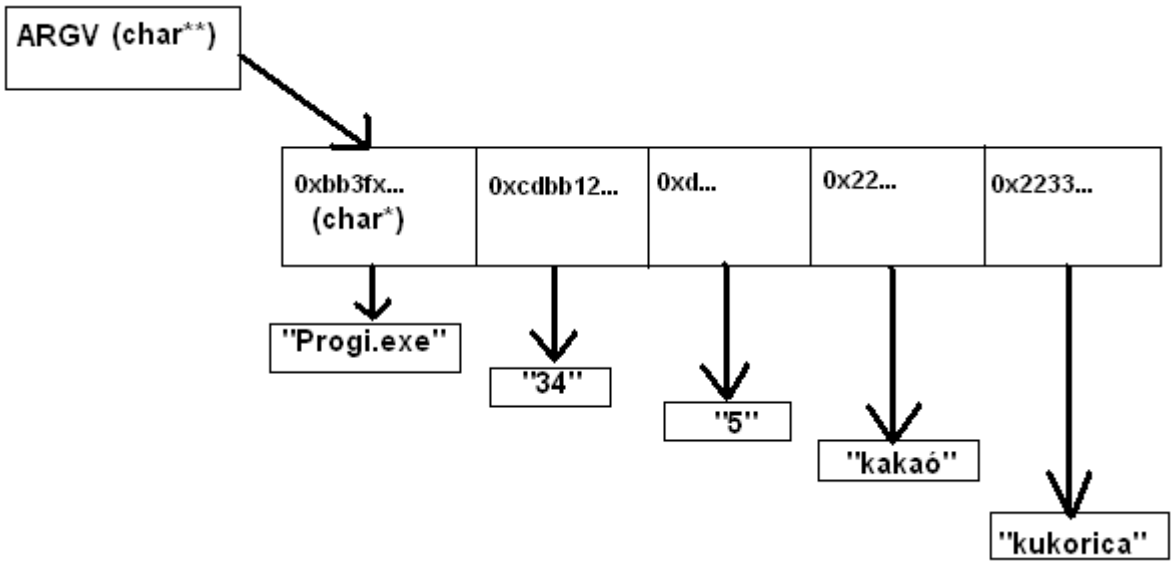
itt argc a beolvasott argumentumok száma ebbe beleszámít maga a „fájlnév.exe” is amit kiadunk a parancssorban.

PÉLDA
(a parancssor pl Futtatás ->cmd paranccsal érhető el MS Windows alatt)

c:\ cd akármi\...\projekt\debug\

c:\akármi\...\projekt\debug\progi.exe 34 5 kakaó kukorica
ekkor argc értéke 5! Mert benne van a progi.exe is!!

Az argv egy olyan pointer ami egy pointer tömbre mutat.



A pointertömb olyan pointereket tárol amik stringekre mutatnak valahol a memóriában. Ezen stringek elérése az argv-n keresztül lehetséges.

1. PÉLDA
Ha a 34-et szeretnénk elérni intiger formátumban azt az argv-n kerszetül a következő képpen tehetjük meg.
argv maga a pointer *argv pedig az az érték ahová a pointer mutat jelen esetbena „0xbb3fx…”

Logikusan ha egyet léptetünk a pointeren és ennek az értékét akarjuk tudni akkor az a következő tömbbeli elem lesz : *(argv+1) ez 0xcdbb12…
Ez ekvivalens azzal mintha az argv-k tömbjében a 2. elemet indexelnénk
*(argv+1)===argv[1]
Megjegyzés : VIGYÁZAT az indexelés 0-tól van

A következő programr kinyomtatja a parancssorra beírt 2. argumentumot.

```
#include<stdio.h>

int main(int argc, char** argv)
{

      printf("%s\n",*(argv+1));

}
```

2. PÉLDA
A következő pedig kinyomtatja az összes parancssorra beirt argumentumot

```
#include<stdio.h>

int main(int argc, char** argv)
{
int i;
for(i=0;i<argc;i++)
{
      printf("%s\n",*(argv+i));//vagy printf("%s\n",argv[i]);//
}
}
```

3.PÉLDA
Ez pedig minden parancssori arg-ot visszafelé ír ki (gyakoroljuk a karakterek elérését)

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

int main(int argc, char** argv)
{
int hossz;              //egy karaktertömb hossza
int i,j;
for(i=0;i<argc;i++)
{
hossz=strlen(argv[i]);
 for(j=hossz-1;j+1;j--)
      {
       printf("%c",argv[i][j]);
      }
 printf("\n");
}
}
```

Innen láthatjuk hogy egy karakter elérése az argv tömb i-dik elemének (ez egy mutató vagyis karaktertömb) j-dik eleme;

ADATOK ELÉRÉSE STRINGBŐL

```
Az
int sscanf(const char *buffer, const char *format, &változó);
```

Függvény segít, hogy kinyerjük az adatokat egy stringből az első paramétere egy karaktertömb azaz egy mutató ami egy karaktertömbre mutat a második paraméter a tárolás formátumát jelöli meg a 3. paraméter pedig a változó ahol tárolni fogjuk az adatot.

Hallgatói Tudásbázis

1.PÉLDA

A következő butácska program demonstrálja például az argv-hez hasonló mutatótömbökből való adatkinyerést.

```
#include<stdio.h>

int main(void)
{
char* e="3.4";
char* g="2.1";
char *a[2];
double *f;
a[0]=e;
a[1]=g;
f=calloc(2,sizeof(double));
sscanf(a[0],"%lf",&f[0]);
sscanf(a[1],"%lf",&f[1]);

printf("%lf\n",f[0]);
printf("%lf\n",f[1]);
}
```

2.PÉLDA

Ez pedig a visual stúdio helpjének példaprogramja

```
#include <stdio.h>

int main( void )
{
   char  tokenstring[] = "15 12 14...";
   char  s[81];
   char  c;
   int   i;
   float fp;

   /* Input various data from tokenstring: */
   sscanf( tokenstring, "%80s", s ); // max 80 character string
   sscanf( tokenstring, "%c", &c );
   sscanf( tokenstring, "%d", &i );
   sscanf( tokenstring, "%f", &fp );

   /* Output the data read */
   printf( "String    = %s\n", s );
   printf( "Character = %c\n", c );
   printf( "Integer:  = %d\n", i );
   printf( "Real:     = %f\n", fp );
}
```

Programozás 2 C++ tárgyból kis- nagyházi segítséget, korrepetálást vállalok vállalok érdeklődés magánban mert nem vagyok a listán

e-mail:cimi333@hotmail.com
Tel.: 06-20-375-23-45
Üdvözlettel : Kristóf