Katedra za elektroniku, Računarska elektronika, 2016

Projekat 14

Prevodjenje M3U fajla u PLS

Projekat radili:

Pajović Aleksandar 414/2013

Adamov Stefan 178/2010

Grupa 13

Program prevodi M3U plejlistu u PLS plejlistu. U M3U formatu, za svaku numeru, prvo se piše vreme, zatim ime pesme i na kraju ime fajla. U PLS je obrnuto, prvo se upisuje ime fajla, zatim ime pesme i na kraju vreme trajanja. Zato je glavna ideja programa da prvo vreme iz M3U formata prepiše na stek, zatim ime pesme, i ime fajla, i da se onda sa steka skida i upisuje u PLS fajl.

Pored toga, PLS format na početku fajla traži ukupan broj numera. To program rešava skanirajuci ceo ulazni fajl i prebrojavajući sve karaktere „#“, i zatim umanjujući taj broj za jedan (zbog prve # koja definiše m3u fajl).

Manipulacija svim tekstom se odvija u memoriji: m3u fajl se prvo učitava u jedan bafer, zatim se koristeći taj bafer popunjava drugi u kojem je tekst za pls fajl, i na kraju se drugi bafer zapisuje u fajl.

Kod:

```
INCLUDE Irvine32.inc
INCLUDE macros.inc

BUFFER_SIZE = 50001
.data
buffer BYTE BUFFER_SIZE DUP(? )
buffer2 BYTE BUFFER_SIZE DUP(? )
srcFilename BYTE 80 DUP(0)
dstFilename BYTE 80 DUP(0)
fileHandle   HANDLE ?
stringLength DWORD ?
bytesWritten DWORD ?
str1 BYTE "Cannot create file", 0dh, 0ah, 0
str2 BYTE "Version=2",0dh, 0ah, 0
const10 dd 10
string4 BYTE "[playlist]", 0dh, 0ah, "NumberOfEntries="
filestring BYTE "File"
titlestring BYTE "Title"
lengthstring BYTE "Length"
BYTE "[Enter]: ", 0dh, 0ah, 0
i byte 1
.code
main PROC


; Let user input a filename.
mWrite "Enter an input filename: "
mov    edx, OFFSET srcFilename
mov    ecx, SIZEOF srcFilename
call   ReadString

;Create an output filname
mov al, '.'
mov ecx, lengthof srcFilename
mov esi, offset srcFilename
mov edi, offset dstFilename
again:movsb
```

```
dec edi
scasb
jne again
mov eax, 'slp'
mov [edi], eax

; Check for errors.
cmp    eax, INVALID_HANDLE_VALUE; error found ?
jne    file1_ok; no: skip
mov    edx, OFFSET str1; display error
call   WriteString
jmp    quit
file1_ok :

sub edx, edx                              ; Open the file for input.
mov    edx,  OFFSET srcFilename
call   OpenInputFile
mov    fileHandle, eax

; Check for errors.
cmp    eax, INVALID_HANDLE_VALUE; error opening file ?
jne    file2_ok; no: skip
mWrite <"Cannot open file", 0dh, 0ah>
jmp    quit; and quit
file2_ok :



; Read the file into a buffer.
mov    edx, OFFSET buffer
mov    ecx, BUFFER_SIZE
call   ReadFromFile
jnc    check_buffer_size; error reading ?
mWrite "Error reading file. "; yes: show error message
call   WriteWindowsMsg
jmp    close_file

check_buffer_size :
cmp    eax, BUFFER_SIZE; buffer large enough ?
jb     buf_size_ok; yes
mWrite <"Error: Buffer too small for the file", 0dh, 0ah>
jmp    quit; and quit

buf_size_ok :
mov    buffer[eax], 0; insert null terminator
mWrite "File size: "
call   WriteDec; display file size
call   Crlf

close_file :
mov    eax, fileHandle
call   CloseFile


cld
;prebrojavanje
mov ebx, 0
mov edi, offset buffer
```

```
mov ecx, lengthof buffer
mov al, '#'
scan: scasb
jne notfound
inc ebx
notfound:
dec ecx
cmp ecx, 0
jne scan

dec ebx
;upis headera
mov edi, OFFSET buffer2
mov esi, OFFSET string4
mov ecx, LENGTHOF string4
rep movsb
call printNumber
inc edi
mov al, 0dh
mov[edi], al
inc edi
mov al, 0ah
mov[edi], al
inc edi

mov esi, OFFSET buffer
mov ebx, 1


; poredjenje sa,
loop1 : mov ecx, 1
       mov al, ","
       cmp [esi], al
       je firstpush
       inc esi
       jmp loop1

       ; upisivanje vremena
       firstpush :
       mov eax, esi
       mov cl, ':'
              firstpushsub:
              cmp [eax], cl
              je loop2
           mov dl, [eax]

              mov [esp], dl
              dec esp
              dec eax
              jmp firstpushsub

loop2:
       mov al, 0dh
       cmp[esi], al
       je secondpush
       inc esi
       jmp loop2
```

```
              ;upisivanje imena
secondpush:
       mov eax, esi
       add esi, 2
       mov cl, ','
       secondpushsub :
                cmp [eax], cl
                je loop3
                mov dl, [eax]
                mov [esp], dl
                dec esp
                dec eax
                jmp secondpushsub

loop3:
       mov al, 0dh
       cmp[esi], al
       je thirdpush
       mov al, 0h
       cmp [esi], al
       je thirdpush

       inc esi
       jmp loop3

              ;upisivanje adrese
thirdpush :
       mov eax, esi
       add esi, 2

       thirdpushsub :
       mov cl, 0ah
       cmp [eax], cl
       je glavniupis
       mov cl, 0h
       cmp [eax], cl
       je specsluc
       mov dl, [eax]
       mov [esp], dl
       dec esp
       dec eax
       jmp thirdpushsub

specsluc: dec eax
dec esi
jmp thirdpushsub


;upis u bafer2
glavniupis:
       mov eax, esi
              mov esi, OFFSET filestring
              mov ecx, LENGTHOF filestring
              rep movsb
              call printNumber
              inc edi
              mov dl, "="
              mov[edi], dl
```

```
                inc edi
                fileloop : inc esp
                mov dl, [esp]
                mov[edi], dl
                inc edi
                cmp dl, 0dh
                jne fileloop
                mov dl, 0ah
                mov[edi], dl
                inc edi
                mov esi, OFFSET titlestring
                mov ecx, LENGTHOF titlestring
                rep movsb
                call printNumber
                inc edi
                mov dl, "="
                mov[edi], dl
                inc edi
                nameloop : inc esp
                mov dl, [esp]
                mov[edi], dl
                inc edi
                cmp dl, 0dh
                jne nameloop
                mov dl, 0ah
                mov[edi], dl
                inc edi
                mov esi, OFFSET lengthstring
                mov ecx, LENGTHOF lengthstring
                rep movsb
                call printNumber
                inc edi
                mov dl, "="
                mov[edi], dl
                inc edi
                timeloop : inc esp
                mov dl, [esp]
                mov[edi], dl
                inc edi
                cmp dl, ","
                jne timeloop
                dec edi
                mov dl, 0dh
                mov[edi], dl
                inc edi
                mov dl, 0ah
                mov[edi], dl
                inc edi
                inc bl
                mov esi, eax
                mov cl, 0h
                cmp[esi], cl
                jne loop1

                ;Upisivanje Version=2, nisam siguran da je ovo potrebno, ali ne skodi
mov eax, esi
mov esi, OFFSET str2
mov ecx, LENGTHOF str2
```

```
rep movsb


; Create a new file.
 xor edx, edx
mov    edx, OFFSET dstFilename
call   CreateOutputFile
mov    fileHandle, eax

; Write the buffer to the output file.
mov    eax, fileHandle
mov    edx, OFFSET buffer2
mov    ecx, LENGTHOF buffer2
call   WriteToFile



quit :
exit


printNumber :
       push eax
       push edx
       xor edx, edx
       mov eax, ebx
       div  const10
       cmp al, 0
       je jedinice
              add al, 30h
              mov[edi], al
              inc edi
jedinice :
              add dl, 30h
              mov[edi], dl
       pop edx
       pop eax
       ret

main ENDP
END main
```