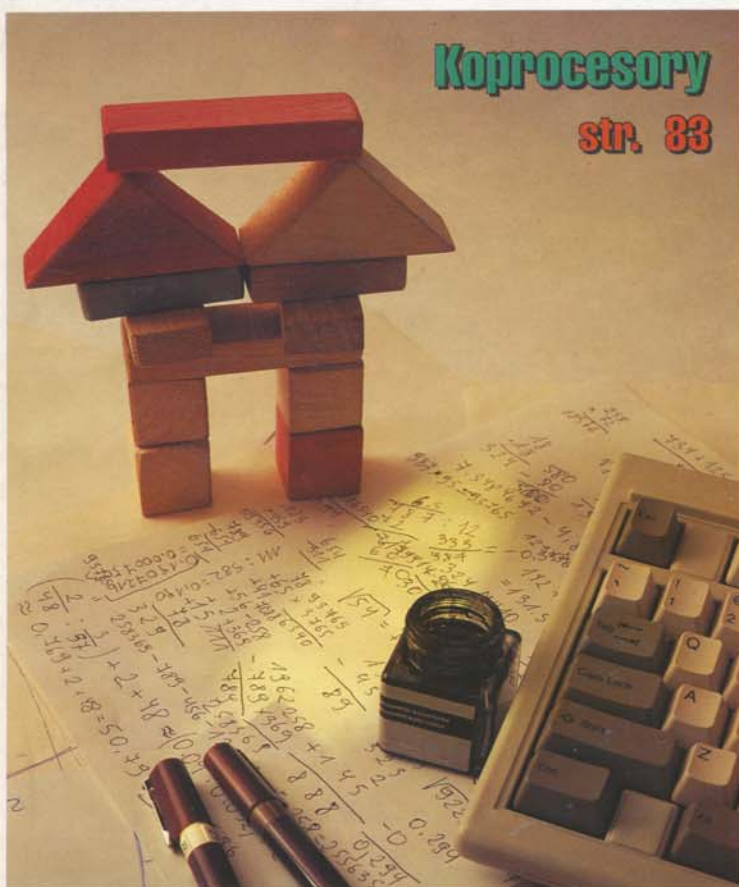


# PCkurier

PISMO UŻYTKOWNIKÓW KOMPUTERÓW OSOBISTYCH



## Windows NT na rynku!

Microsoft skierował wreszcie do produkcji Windows NT i w najbliższych dniach program ten powinien się pojawić w sieci sprzedaży. Użytkownik dostaje do ręki system operacyjny nowej generacji, obudowany ponad 2 tysiącami gotowych, 32-bitowych aplikacji, które równolegle wchodzi na rynek. Czołowe firmy komputerowe zapowiedziały preinstalowanie systemu na dostarczanych przez siebie maszynach (m.in. Acer, Compaq, Dell, ICL, NCR, Olivetti, SNI). Obecna realizacja obejmuje wersje dla procesorów Intel (od 386) i MIPS R4400, natomiast w ciągu najbliższych dwóch

miesięcy będzie gotowa wersja dla procesora Alpha AXP.

## Corel kupuje Venturę?

Corel Corp. podała pod koniec sierpnia informację o rozpoczęciu starań o zakupienie od Ventura Software technologii programów Ventura Publisher, Ventura Database Publisher oraz oprogramowania towarzyszącego. Kontrakt ma zostać sfinalizowany do końca września. Wstępne rozmowy zakładają, że Corel zakupi także wersje dla różnych platform i wersje zlokalizowane Ventury oraz pełny kod i prawa do bazy klientów Ventury.

## Raz, dwa, trzy... 4.0

Mirosław Kalfiński

*Lotus kontynuuje ofensywę przeciwko swemu najgroźniejszemu konkurentowi w dziedzinie arkuszy kalkulacyjnych - Excelowi Microsofta. Odpowiedzią na Excela 4.0 ma być ulepszona, przeznaczona specjalnie dla Windows, wersja 4.0 arkusza 1-2-3.*

Lotus startuje z pozycji, w której musi (według mnie) doganiać osiągnięcia Excela, wprowadził więc do 1-2-3 ulepszenia, które przekonają mają użytkowników do tego produktu.

Spodziewałem się, że 1-2-3 będzie podobnie jak inne produkty Lotus'a - bardzo starannie dopracowany graficznie i użytkowo, ale bardzo wymagający dla użytkownika. To się potwierdziło. Formalne wymagania są wprawdzie typowe dla rozbudowanych programów dla Windows: DOS w wersji co najmniej 3.30, Windows 3.1 (może być 3.0), 4 MB RAM, 13 MB na dysku dla zainstalowania wszystkich opcji (+5 MB, jeśli ktoś chce obejrzeć sobie ładne animowane wprowadzenie - Guided Tour, które Lotus dodaje do wszystkich już chyba produktów).

Obecnie dla "sensownej" pracy w Windows zalecane jest posiadanie 4 MB RAM, formalnie wymaga się przynajmniej 2 MB, więc użytkownicy słabszych komputerów nie będą po prostu w stanie skorzystać z nowego 1-2-3 (nawet gdy z powodzeniem uruchomią Windows).

(c.d. na str. 115)

W następnym numerze:

- Programowanie w C++
- Windows NT

Jedynym sensownym wyjściem, zalecanym zresztą i przy innych okazjach, jest porządne podawanie parametrów. Np. dla tak zadeklarowanej zmiennej:

```
var MyDear : array[1..17]
  of record
    HisDear : array [1..4]
              of string[80];
    First,
    Second : string[23];
  end;
```

wywołanie funkcji pPolBuf powinno wyglądać następująco:

```
pPolBuf(pILatin, pINPL,
  Seg(MyDear), Of(MyDear), SizeOf(MyDear));
```

Niedogodność tę rekompensuje nam działanie procedury. Możemy na przykład dokonać zmiany standardu polskich znaków w pamięci ekranu lub zmienić polskie litery w całym buforze tekstowym, jeśli taki sobie odpowiednio stworzymy (w module JZKBuf). Możemy wreszcie napisać wspomniany program JZKConv, o banalnym algorytmie działania:

```
while not eof(przetwarzany_plik) do begin
  wczytaj_kawalek_pliku;
  przetwórz_wczytany_kawalek_pliku;
  zapisz_wczytany_kawalek_pliku; end;
```

Przy okazji uczulam na drobną sprawę. Jeżeli którykolwiek z elementów zmiennej MyDear będzie ciągiem o zadeklarowanej długości większej niż 127, możemy spodziewać się drobnych błędów. Pierwszym bowiem znakiem każdego turbopaskalowego ciągu jest Chr(L) (L jest długością owego ciągu), czyli także będzie poddany konwersji.

Fanatycy elegancji (do których czasem też się zaliczam) zechcą zapewne dorzucić, jeszcze w części inicjalizującej modułu, pewną procedurę. Powinna ona zorientować się, czy któraś ze zmiennych otoczenia nie ma przypadkiem nazwy PL i czy aby nie odpowiada jej jakaś sensowna wartość. Jeżeli tak, należy przypisać zmiennej pLStandard odpowiednią stałą. Tak samo należy następnie potraktować parametry przekazane przez COMMAND.COM w czasie uruchamiania naszego procesu. Tu, naszym znakiem rozpoznawczym powinna być opcja "/PL=" lub "/PL=" albo coś w tym stylu. I już możemy dostać od PCMAGAZINE PO POLSKU, przy okazji oceny naszego programu, dodatkowe punkty za elegancką współpracę z polskimi literami.

To tyle. Chciałbym kiedyś dożyć czasów, kiedy podobne narzędzia dostaniemy od razu wraz z systemem operacyjnym i nie trzeba będzie bawić się w tworzenie własnych. Choć jest to naprawdę wysmienita zabawa na lato.

Kontynuując cykl "czekolada dla każdego", proponuję mały konkurs. Trzy duże wedlowskie czekolady z orzechami rozlosuję wśród osób, które nadesłały odpowiedź na pytanie, dlaczego procedura pPolBuf nie może zostać uproszczona do postaci o nagłówku:

```
pPolBuf(InStd, OutStd : byte; var Buffer);
```

Na odpowiedzi czekam przez najbliższe dwa tygodnie pod adresem: Jarosław Z. Knypl, studioJZK, ul. Skarpowa 98/2, 80-140 Gdańsk-Siedlce.

Ekstra czekolada dla osoby, która rozwikła zagadkę - jak dobrać się do tablicy konwersji, która jest tworzona przez system operacyjny po deklaracji typu:

```
COUNTRY=048,852,C:\DOS\COUNTRY.SYS
```

umieszczonej w pliku CONFIG.SYS. □

## Funkcje obsługi ekranu wykorzystujące ANSL.SYS

Jacek Kuras

Wraz z systemem DOS dostarczany jest sterownik ANSL.SYS (ang. device driver). Oferuje on użytkownikom mikrokomputerów typu IBM PC pewien zestaw funkcji obsługi ekranu i klawiatury. Wywołanie odpowiedniej funkcji sterownika polega na wypisaniu na konsoli ciągu znaków (charakterystycznych dla tej funkcji) zaczynającego się zawsze znakami Esc [. Znaki te nie pojawiają się na ekranie monitora. Funkcje sterownika ANSL.SYS mogą być wywoływane tylko wtedy, gdy zostanie on zainstalowany w trakcie startu systemu. Uzyskujemy to przez umieszczenie w pliku CONFIG.SYS linii postaci: DEVICE=C:\DOS\ANSL.SYS (jeżeli plik ANSL.SYS znajduje się w katalogu C:\DOS) lub podobnej.

Chciałbym przedstawić Czytelnikom PCkuriera pewien zestaw pożytecznych podprogramów obsługi ekranu, wykorzystujących sterownik ANSL.SYS. Każdy z nich jest bardzo prosty i zawiera tylko jedną instrukcję wypisującą odpowiedni ciąg znaków. Podprogramy zostały napisane w języku Pascal, ale wzorując się na nich możemy w ciągu kilkunastu minut napisać analogiczne podprogramy w dowolnym, innym języku. Użytkownicy Turbo Pascala zauważą zapewne, że niektóre z tych podprogramów mają swoje odpowiedniki w bibliotece tego języka. Są jednak kompilatory języków programowania, które nie zawierają w swoich bibliotekach analogicznych podprogramów. Ta-

kim językiem jest na przykład FORTRAN, używany przez niewielką grupę programistów. Pisząc podprogramy w tym języku należy pamiętać o poprzedzeniu w formacie ciągu znaków - znakiem spacji (w FORTRAN-ie pierwszy znak jest zawsze traktowany jako znak sterujący). Podobne, niewielkie modyfikacje mogą być niezbędne w innych językach programowania. Przedstawione podprogramy nie wyczerpują wszystkich możliwości oferowanych przez sterownik ANSL.SYS. Bardziej ambitnych Czytelników zachęcam do dokładniejszego zapoznania się z jego funkcjami i zaprojektowania własnych podprogramów, np. ustawiających odpowiednie atrybuty dla wyświetlanego na ekranie tekstu (kolor znaku, kolor tła, migotanie itp.). Wyczerpującą informację o funkcjach oferowanych przez sterownik ANSL.SYS znaleźć można w podanej literaturze, a także w wielu innych opracowaniach dotyczących systemu operacyjnego DOS. Użytkownicy systemu MS DOS 6.0 pełną informację uzyskają na monitorze komputera przez wywołanie polecenia HELP ANSL.SYS. □

### Literatura:

- [1] Microsoft MS-DOS. User's Guide and Reference (dokumentacja dostarczana z systemem MS DOS 5.0)
- [2] Daniel Rouge, MS-DOS pomysły i zastosowania
- [3] Bulhak J., Gończyński R., Tuszyński M., DOS 5.00 od środka

```

unit ANSIScrn; (
| Zestaw podprogramow wykorzystujacych |
| sterownik ANSI.SYS do obslugi ekranu |
| w trybie tekstowym |
| (c) Jacek Kuras |
)
interface
  procedure ANSI_Init;
  procedure ANSI_ClrEol;
  procedure ANSI_GotoXY (x, y: integer);
  procedure ANSI_ClrScr;
  procedure ANSI_CursorUp (n: integer);
  procedure ANSI_CursorDown (n: integer);
  procedure ANSI_Forward (n: integer);
  procedure ANSI_Backward (n: integer);
  procedure ANSI_SaveCursorPosition;
  procedure ANSI_RestoreCursorPosition;
implementation
const
  Esc_ = chr(27) + '[';
var
  console: text;
  procedure ANSI_Init; (
  | Zainicjuj modul "ANSIScrn" |
  )
  begin
    assign (console, 'CON');
    rewrite (console);
    end; ( ANSI_Init )
  procedure ANSI_ClrEol; (
  | Usun wszystkie znaki od aktualnego |
  | polozenia kursora do konca wiersza |
  )
  begin
    write (console, Esc_, 'K')
  end; ( ANSI_ClrEol )
  procedure ANSI_GotoXY; (
  | Przesun kursor w polozenie o wspolrzadnych (x,y) |
  )
  begin
    write (console, Esc_, y, ';', x, 'H')
  end; ( ANSI_GotoXY )
  procedure ANSI_ClrScr; (
  | "Wyczysc" ekran |
  )
  begin
    write (console, Esc_, '2J')
  end; ( ANSI_ClrScr )
  procedure ANSI_CursorUp; (
  | Przesun kursor o "n" wierszy w gore |
  | nie zmieniajac aktualnej kolumny |
  )
  begin
    write (console, Esc_, n, 'A')
  end; ( ANSI_CursorUp )
  procedure ANSI_CursorDown; (

```

```

  | Przesun kursor o "n" wierszy w dol |
  | nie zmieniajac aktualnej kolumny |
  )
  begin
    write (console, Esc_, n, 'B')
  end; ( ANSI_CursorDown )
  procedure ANSI_Forward; (
  | Przesun kursor o "n" kolumn w prawo |
  | nie zmieniajac aktualnego wiersza |
  )
  begin
    write (console, Esc_, n, 'C')
  end; ( ANSI_Forward )
  procedure ANSI_Backward; (
  | Przesun kursor o "n" kolumn w lewo |
  | nie zmieniajac aktualnego wiersza |
  )
  begin
    write (console, Esc_, n, 'D')
  end; ( ANSI_Backward )
  procedure ANSI_SaveCursorPosition; (
  | Zapamietaj aktualne polozenie kursora |
  )
  begin
    write (console, Esc_, 's')
  end; ( ANSI_SaveCursorPosition )
  procedure ANSI_RestoreCursorPosition; (
  | Przesun kursor w polozenie zapamietane przy |
  | pomocy procedury ANSI_SaveCursorPosition |
  )
  begin
    write (console, Esc_, 'u')
  end; ( ANSI_RestoreCursorPosition )
end. ( ANSIScrn )

```

**PROGRAMATORY EPROM**

EPW 2048 do 2MB      cena 1700000 zł

**SYMULATORY EPROM**

SE 2000 do 2MB      cena 1400000 zł

**MODEMY TM 1200**

IBM, Amiga, Atari ST, homolog.      cena 500000 zł

**AUTOMATYCZNY MONTAŻ SMD**

LARS Co

02-795 Warszawa, ul. Kazury 28,  
tel/fax 40-63-34**SOFT**  
design**KOMPLETNE STANOWISKO GRAFICZNE  
TYLKO ZA... 180 mln. (cena zawiera VAT)**tel./fax 37-37-14, 37-05-65  
tel. 37-80-00, 32-14-91 w. 274, 275  
01-164 Warszawa, ul. Radziwie 13PC 486/33, 16 MB RAM, hdd 245 MB  
monitor kolor SVGA, mysz  
skaner Microtek ScanMaker II  
drukarka postscriptowa NewGen PS 300p  
DOS 6.0, Windows 3.1, CorelDRAW! 3.0PL  
PhotoShop, Ventura Publisher**KOMPUTERY I POLIGRAFIA**