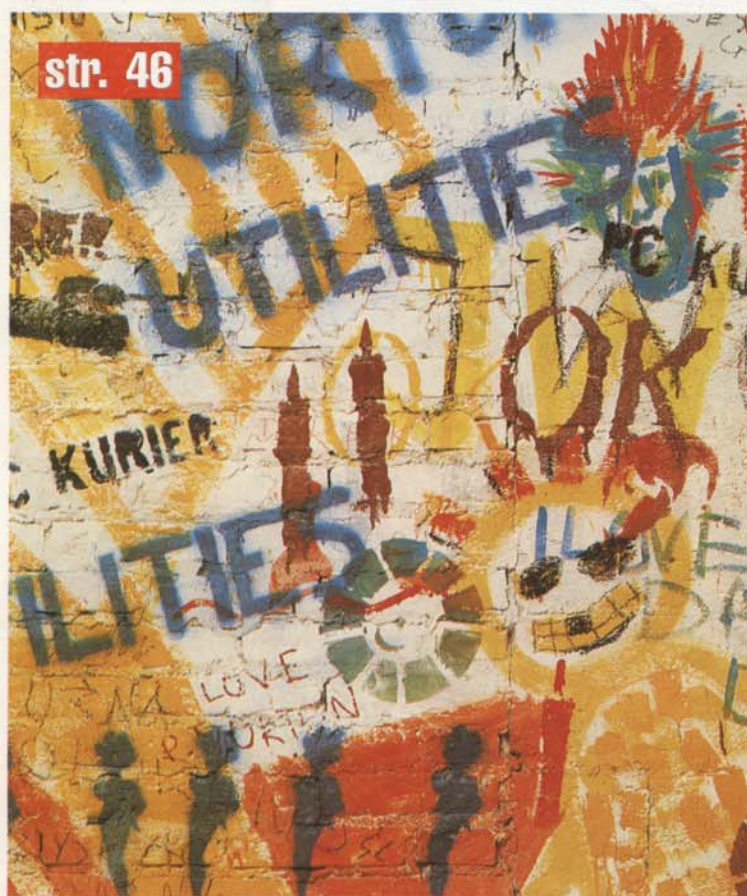


# PCkurier

PISMO UŻYTKOWNIKÓW KOMPUTERÓW OSOBISTYCH



str. 46

## Pierwszy model Cray-3

Znana firma *Cray Computer Corp.*, mająca swoją siedzibę w Colorado Springs, wypuściła wreszcie pierwszy egzemplarz najnowszego superkomputera Cray-3. Użytkownikiem tej najszybszej w przemyśle komputerowym maszyny będzie National Center for Atmospheric Research (Państwowy Ośrodek Badań Atmosferycznych), który nic nie będzie za nią płacił w ciągu sześciu miesięcy okresu próbnego. Superkomputer posiada 4 procesory, 128 M-słów i pracuje z zegarem taktującym o częstotliwości 480 MHz.

JD 14011

## Ingres szykuje bazę danych

Firma *Ingres Corp.* postanowiła ujawnić wkrótce wersję swojej bazy danych w postaci NetWare Loadable Module - wiadomość taką przynosi pismo *Computer Resellers News*. Wersja dla NetWare 3.X spodziewana jest w końcu roku, zaś NetWare 4.0 Loadable Module pojawi się na początku 1994 roku. Wersja dla pojedynczego użytkownika, bazująca na NetWare Runtime 3.11, ma kosztować 750 dolarów, zaś wersje dla serwerów będą kosztowały od 3900 dolarów dla ośmiu użytkowników, aż do 27 000 dolarów dla 64 użytkowników.

JD 14008

## Ergonomia a komputery

Tomasz Kulisiewicz

*Światowa Konferencja Międzynarodowego Stowarzyszenia Ergonomicznego, która odbyła się od 14 do 17 czerwca w Warszawie, poświęcona ergonomii "powtarzalnych prac ręcznych oraz zautomatyzowanych", a także przetwarzaniu informacji w pracy, stała się okazją przedstawienia problemu ergonomii pracy z komputerami.*

Pojęcia "ergonomia" dla określenia przystosowania stanowiska pracy do psychofizycznych możliwości człowieka użył po raz pierwszy Polak, Wojciech Jastrzębowski, w roku 1857. Jako dyscyplina naukowa - integrująca wiele dziedzin wiedzy - ergonomia rozwinęła się w okresie II wojny światowej, kiedy to okazało się, że w złożonych systemach militarno-technicznych najsłabszym ogniwem jest człowiek, a ściślej: słabe dostosowanie tych systemów do możliwości psychofizycznych człowieka, który miał nad nimi panować.

Zaczęto wtedy poszukiwać przyczyn takiego stanu rzeczy oraz metod zapobiegania "niedostosowaniu człowieka do maszyn". Maszyny zmieniać łatwiej, człowieka trudniej. Choć za pomocą różnych środków - głównie szkoleń i treningów - próbowano zmieniać i człowieka, to jednak było oczywiste, iż żaden trening nie zmieni faktu, że rąk mamy tylko dwoje, że sięgamy nimi na określoną odległość, zaś przedmioty widzimy lepiej, kiedy są dobrze oświetlone.

(c.d. na str. 43)

W następnych numerach:

- dBase IV 2.0
- Visual Basic

## Niejednoznaczności obsługi klawiatury - ciąg dalszy

Jacek Kuras

Wieloletnia praktyka w nauczaniu programowania spowodowała, że do wszelkich informacji o błędach kompilatorów podchodzę bardzo ostrożnie. Usterki przypisywane kompilatorom przez programistów (szczególnie mało doświadczonych) są najczęściej efektem błędu autora, pełnionym w trakcie projektowania algorytmu, a nie rzadko źródłem ich można szukać w niedokładnym zapoznaniu się przez programistę z dokumentacją użytkową. Wszystko to powoduje, że bardzo szczegółowo analizuję artykuły, w których autorzy sygnalizują takie błędy.

Podobnie postąpiłem z artykułem p. Piotra Kowalskiego - "Niejednoznaczności obsługi klawiatury" (PCKurier 11/93). Przytoczone przez Autora spostrzeżenia potwierdziły się. Nie można jednak w pełni twierdząc odpowiedzieć na postawione w końcu artykułu pytanie: "Czyżby niedopatrzenie firmy Borland?". W dokumentacji Turbo Pascala 6.0 (Programmer's Guide str. 350) przedstawiona została tabela kodów klawiszy specjalnych i ich kombinacji, generujących podwójny kod dla funkcji ReadKey (zero i kod właściwy). W tabeli tej nie ma kombinacji Ctrl-Ins i Ctrl-Del, z czego można wnioskować, że nie przewidziano obsługi tych klawiszy w funkcjach KeyPressed i ReadKey, a efekt uzyskany w środowisku Turbo Pascala jest uboczny (w środowisku najnowszego Turbo Pascala 7.0 Ctrl-Ins i Ctrl-Del nie są już obsługiwane przez te funkcje).

Przedstawione w artykule "Niejednoznaczności obsługi klawiatury" funkcje WHATKEY i IFKEY spełniają założone przez Autora cele, ale są zbyt złożone. Projektując algorytmy programów, powinniśmy zawsze starać się, aby były one jak najprostsze (dzięki temu program staje się elegantski i przejrzysty).

W tym artykule pragnę zaprezentować inną, o wiele prostszą realizację funkcji, odczytujących klawiaturę, a ponadto dysponujących większymi możliwościami. Język Turbo Pascal umożliwia programiście dostęp do przerwań BIOS i DOS (podprogram Intr). Komunikacja z klawiaturą mikrokomputerów typu IBM PC obsługiwana jest za pomocą przerwania 16H ROM BIOS. Wywołanie funkcji 0 tego przerwania powoduje odczytanie znaku i usunięcie go z buforu klawiatury. Funkcja 1 umożliwia testowanie stanu klawiatury, a także odczytanie znaku bez usuwania go z buforu. Za pomocą funkcji 2 można sprawdzać aktualny stan klawiszy: Insert, CapsLock, NumLock, ScrollLock, Alt, Ctrl i Shift. Przed wywołaniem w Pascalu podprogramu Intr, numer funkcji dla przerwania 16H należy wpisać do zmiennej, reprezentującej rejestr AH.

Wraz z pojawieniem się mikrokomputerów IBM PC/AT i rozszerzonej klawiatury (101 klawiszy) przerwanie 16H wzbogacone zostało o nowe funkcje o numerach: 10H, 11H i 12H, będące odpowiednio rozszerzeniami funkcji 0, 1 i 2. Zapewne wiele osób próbowało bezskutecznie odczytać klawisze F11 i F12 rozszerzonej klawiatury procedurą ReadKey. Nie można też odczytać tych klawiszy funkcją WHATKEY. Zaprezentowane poniżej bardzo proste odpo-

wiedniki funkcji ReadKey i KeyPressed zrealizowane zostały w oparciu o przerwanie 16H ROM BIOS. Funkcje ReadKeyCode i KeyPressed\_ reagują dodatkowo na co najmniej 20 różnych kombinacji klawiszy, których nie odczytują zarówno ReadKey i KeyPressed, jak i WHATKEY i IFKEY. Są to klawisze F11 i F12 (wraz z kombinacjami z Shift, Ctrl i Alt), oraz kombinacje klawiszy Ctrl lub Alt z niektórymi klawiszami klawiatury numerycznej, klawiszem Tab lub klawiszem Backspace.

Podprogramy te nie będą działać na wszystkich komputerach XT oraz komputerach AT ze starymi wersjami BIOS. Dla tych komputerów należy w funkcji ReadKeyCode zastąpić \$10 przez \$0, a w funkcji KeyPressed\_ \$11 zastąpić przez \$1. Po tych modyfikacjach funkcje te będą działały analogicznie, jak pascalowe KeyPressed i ReadKey zarówno w środowisku Turbo Pascala, jak i w DOS. Możemy więc wysnuć z tego faktu wniosek, że funkcje KeyPressed i ReadKey zostały zrealizowane w oparciu o funkcje 0 i 1 przerwania 16H. Zrealizowane je w ten sposób, aby działały poprawnie na wszystkich typach komputerów. Stąd właśnie wynikają pewne ich ograniczenia. Widzimy więc, że odmienne działanie funkcji KeyPressed i ReadKey w środowisku Turbo Pascala i w DOS jest po prostu efektem własnej obsługi przerwania 16H w środowisku Turbo Pascala (co w wersji 7.0 zostało już usunięte).

Na zakończenie pragnąłbym zwrócić uwagę na fakt, iż funkcja ReadKeyCode jest wygodniejsza w użyciu niż klasyczna funkcja ReadKey, którą należy wywoływać dwukrotnie, gdy naciśnięty został klawisz specjalny. Funkcję ReadKeyCode wywołujemy zawsze tylko jeden raz, otrzymując kod znaku dla klawiszy znakowych, a dla klawiszy specjalnych ujemną wartość Scan Code (kod klawisza lub kombinacji klawiszy).

### Literatura:

Turbo Pascal 6.0, Programmer's Guide.

Turbo Pascal 6.0, Library Reference.

Ray Duncan, IBM ROM BIOS.

```

program test;
-----
| Rozszerzona wersja funkcji ReadKey |
| i KeyPressed |
| (c) Jacek Kuras |
-----
uses Crt, Dos;
var
  i: integer;
function ReadKeyCode: integer;
-----
| Odczyt znaku z buforu klawiatury. Funkcja |
| zwraca kod znaku (dla klawiszy specjalnych |
| ujemny Scan Code) |
-----
var
  regs: Registers;

```