

Termin zaliczania 4-5 tydzień

Zadanie 1.

Wykorzystując dowolny język programowania dla komputerów w standardzie PC napisać aplikację wykorzystującą technikę tworzenia graficznego interfejsu użytkownika. Interfejs powinien w sposób interaktywny wykorzystywać do komunikacji z operatorem monitor ekranowy, klawiaturę oraz myszkę.

Wykorzystanie interfejsu może być dowolne. Sugerowane aplikacje to prosta gra interaktywna lub kalkulator z zegarem. W obu przypadkach wskazane jest aby elementy grafiki były w pewnym stopniu konfigurowane przez użytkownika np. wybór „skórki” dla kalkulatora oraz kolorów i rodzaju wyświetlania czasu (zegar analogowy lub cyfrowy).

Opracowanemu programowi powinna towarzyszyć dokumentacja – sprawozdanie. Sprawozdanie powinno zawierać:

- *sformułowanie zadania wraz z przyjęciem założeń szczegółowych np. algorytm gry, opis elementów jej grafiki;*
- *opis przyjętych rozwiązań programowych zilustrowanych ewentualnie fragmentami kodu (nie zamieszczać wydruków całych programów!);*
- *dyskusję osiągniętych wyników z wskazaniem wad i zalet napisanej aplikacji.*

Termin zaliczania 6-7 tydzień

Zadanie 2.

Tematem zadania czwartego jest wykorzystanie interfejsów dostępnych w komputerze PC do komunikacji pomiędzy dwoma komputerami. Wykorzystując dowolnie wybrany język programowania napisać aplikację symulującą transmisję szeregową zgodną ze standardem RS232. W oknie nadajnika przygotowujemy tekst do nadania w postaci ciągu znaków ASCII. Następnie tekst ten jest formatowany do formatu bit startu, bity znaku od LSB do MSB i dwa bity stopu i zapisywany do łańcucha (każdy znak ASCII posiada własny bit startu i bity stopu!!!). Łańcuch powinien zostać wyświetlony w osobnym polu na ekranie. Następnie łańcuch jest przesyłany do odbiornika, którym może być drugie okno tego samego programu, drugi program lub nawet inne pole panelu tego samego programu. Jako nośnika danych można użyć: tablicy (gdy nadajnik i odbiornik są w tym samym programie), pliku lub wybranego protokołu komunikacyjnego np. TCP/IP – przy czym pierwsze dwie metody są wystarczające. W odbiorniku łańcuch danych zostaje poddany dekodowaniu, tj. usunięciu ramek i dekompozycji szeregowo równoległej. Tekst po zamianie na ciąg znaków ASCII powinien zostać wyświetlony w oddzielnym polu. Elementem aplikacji powinna być również dbałość o czystość języka. Należy to zrealizować w postaci słownika „grubiaństw” zawartego w osobnym pliku. Napotkanie w nadawanym lub odbieranym tekście „grubego” słowa ze słownika powinno skutkować zastąpieniem wszystkich jego liter ciągiem gwiazdek.

Opracowanemu programowi powinna towarzyszyć dokumentacja – sprawozdanie. Sprawozdanie powinno zawierać:

- *sformułowanie zadania wraz z przyjęciem założeń szczegółowych np. zasad przekazywania informacji pomiędzy nadajnikiem i odbiornikiem, konwersji znak ASCII <-> strumień bitów, mechanizmów stosowania filtracji „niecenzuralnych” słów;*
- *opis przyjętych rozwiązań programowych zilustrowanych ewentualnie fragmentami kodu (nie zamieszczać wydruków całych programów!);*
- *dyskusję osiągniętych wyników z wskazaniem wad i zalet napisanej aplikacji.*

Termin zaliczania 8-9 tydzień

Zadanie 3.

Tematem zadania piątego jest napisanie aplikacji uzależnionej od czasu. Jednym z zadań aplikacji ma być możliwie precyzyjny pomiar lub odmierzanie czasu. Wykorzystując dowolny język programowania dla komputerów w standardzie PC napisać aplikację spełniającą funkcję testera sprawności psychomotorycznej np. kandydatów na kierowców. Na aplikację powinna się składać seria różnych testów badających prosty i złożony czas reakcji na bodźce optyczne i akustyczne. Każdy test właściwy powinna poprzedzać informacja o przebiegu testu oraz faza szkoleniowa, w trakcie której osoba badana wykona te same czynności co w trakcie testu, ale bez oceny. Po wykonaniu serii testów osoba poddana badaniom powinien zostać poinformowana o osiągniętych wynikach w formie syntetycznej i analitycznej z wykorzystaniem wartości liczbowych i reprezentacji graficznej.

Opracowanemu programowi powinna towarzyszyć dokumentacja – sprawozdanie. Sprawozdanie powinno zawierać:

- *sformułowanie zadania wraz z przyjęciem założeń szczegółowych np. sposobu pomiaru czasu i określania interwałów czasowych;*
- *opis przyjętych rozwiązań programowych zilustrowanych ewentualnie fragmentami kodu (nie zamieszczać wydruków całych programów!);*
- *dyskusję osiągniętych wyników z wskazaniem wad i zalet napisanej aplikacji.*

Termin zaliczania 10-11 tydzień

Zadanie 4.

Tematem zadania szóstego jest napisanie aplikacji, będącej symulatorem stanowiska dyspozytorskiego „linii produkcyjnej”. Program ten ma zawierać elementy diagnostyki nadzorowanego „procesu produkcji” jak i autodiagnostyki operatora czuwającego nad prawidłowym przebiegiem „produkcji”. Wykorzystując dostępne informacje na temat parametrów pracy komputera PC (np. temperatury rdzenia procesora, stopnia wykorzystania procesora, prędkości obrotowych wentylatorów itp.) oraz generatory liczb losowych i timery, zasymulować parametry kontrolowanego „procesu produkcyjnego”. Należy przewidzieć obsługę pojawiających się losowo awarii oraz przekroczeń granicznych wartości wybranych parametrów procesu - np. po przekroczeniu granicznej temperatury obudowy silnika należy włączyć dodatkowy wentylator lub zwolnić tempo pracy linii produkcyjnej itp. O wszystkich wyjątkowych zdarzeniach i wymaganych działaniach, operator musi być informowany za pośrednictwem odpowiedniego zestawu

komunikatów. Program powinien zawierać okno logowania do aplikacji i na bieżąco badać obecność oraz „przytomność” operatora. Element autodiagnostyczny powinien polegać na okresowym pojawianiu się komunikatu informującego o konieczności potwierdzenia obecności przez wciśnięcie wybranego klawisza. W przypadku braku potwierdzenia, np. przez co najmniej 30 sekund, powinno nastąpić uruchomienie alarmu i wylogowanie operatora z systemu.

- Opracowanemu programowi powinna towarzyszyć dokumentacja – sprawozdanie. Sprawozdanie powinno zawierać:
 - sformułowanie zadania wraz z przyjęciem założeń szczegółowych np. sposobu badania, wykrycia i możliwości zasymulowania awarii;
 - opis przyjętych rozwiązań programowych zilustrowanych ewentualnie fragmentami kodu (nie zamieszczać wydruków całych programów!);
 - dyskusję osiągniętych wyników z wskazaniem wad i zalet napisanej aplikacji.

Termin zaliczania 12-13 tydzień

Zadanie 5.

Wykorzystując dowolny język programowania dla komputerów w standardzie PC napisać aplikację stanowiącą model programowego symulatora mikroprocesora. Program powinien mieć przyjazny interfejs użytkownika graficzny lub znakowy do uznania przez autorów. Należy przyjąć następujący model naszego procesora:

- Procesor posiada cztery 16-bitowe rejestry ogólnego przeznaczenia oznaczone jako AX, BX, CX i DX;
- Każdy z rejestrów powinien być traktowany jako para 8-bitowych rejestrów o oznaczeniach NH dla części starszej i NL dla części młodszej gdzie N oznacza A albo B albo C albo D;
- Lista rozkazów naszego procesora obejmuje trzy rozkazy MOV – przesłania, ADD – dodawania i SUB – odejmowania;
- Procesor umożliwia realizację dwóch trybów adresowania: trybu rejestrowego oraz trybu natychmiastowego;
- Program umożliwia pisanie krótkich programów z użyciem dostępnych rozkazów i trybów adresowania;
- Program umożliwia realizację napisanych programów w trybie całościowego wykonania i w trybie pracy krokowej.
- W trybie pracy krokowej należy zapewnić śledzenie kolejności wykonywanych instrukcji. Zalecenie to należy zrealizować stosując numerację linii kodu programu oraz odwołania do numeru aktualnie wykonywanej instrukcji.
- Program symulatora umożliwia zapisanie do pliku a następnie ponowne wczytanie napisanego wcześniej programu w celu jego dalszej edycji i ponownego uruchomienia.

- Opracowanemu programowi powinna towarzyszyć dokumentacja – sprawozdanie. Sprawozdanie powinno zawierać:
 - Sformułowanie zadania wraz z przyjęciem założeń szczegółowych np. w zakresie stosowanej składni, separatorów kodów mnemotechnicznych i argumentów itd.;
 - Opis przyjętych rozwiązań programowych zilustrowanych ewentualnie fragmentami kodu (nie zamieszczać wydruków całych programów!) dotyczących kluczowych kwestii programu np. parsing linii programu;
 - Dyskusję osiągniętych wyników z wskazaniem wad i zalet napisanej aplikacji.

Termin zaliczania 14-15 tydzień

Zadanie 6.

Do wyboru temat A lub B:

A. Temat dowolny.

B. Rozszerzyć możliwości modelu programowego symulatora mikroprocesora opracowanego w zadaniu 1 tak aby umożliwiał on symulację realizacji dziesięciu funkcji przerwań procesora oferowanych przez moduł BIOS (można też użyć przerwań DOS).

Wybrane funkcje powinny być różnorodne, dotyczyć różnych zasobów komputera PC np. zegara RTC, klawiatury, monitora ekranowego, pamięci dyskowej. Aplikacja powinna realizować dwa wątki:

- dydaktyczny – prezentujący w wyczerpujący sposób opis działania funkcji oraz sposób jej programowego wykorzystania obejmujący sposób wywoływania oraz przekazywania parametrów oraz wyjaśniający znaczenie tych parametrów.
- demonstracyjny – prezentujący wybrane funkcje w działaniu.

W celu realizacji wątku demonstracyjnego do zbioru nazw dotychczas realizowanych komend (MOV, ADD, SUB) należy dodać INTxx, INTyy, INTzz, , gdzie xx, yy, zz, oznaczają numery wybranych przerwań. W programie należy utworzyć STOS służący do przechowywania zawartości rejestrów procesora na czas wykonywania przerwań. (Szczegółowy opis mechanizmu wywoływania przerwań procesora zostanie podany przez prowadzącego w odrębnym załączniku.)

- Opracowanemu programowi powinna towarzyszyć dokumentacja – sprawozdanie. Sprawozdanie powinno zawierać:
 - sformułowanie zadania wraz z przyjęciem założeń szczegółowych np. opisu wybranych funkcji i sposobu ich wykorzystania itd.;
 - opis przyjętych rozwiązań programowych zilustrowanych ewentualnie fragmentami kodu (nie zamieszczać wydruków całych programów!);
 - dyskusję osiągniętych wyników z wskazaniem wad i zalet napisanej aplikacji.