|  |  |
| --- | --- |
| **1. Korzyści korzystania z Makra**  \*redukcja liczby błędów popełnianych przez programistę  \*Umożliwa zdefiniowanie często używanych sekwencji instrukcji  \*Wielokrotne użycie tego samego makra w tekscie programu źrodlowego każdorazowo zapewni ten sam efekt.  \*Skraca czas przygotowania programu źródłowego, zyskuje on na przejrzystości  \*Symbole, etykiety uzyte w makro są lokalne w obrebie makro  **2. Roznice między HLL a ASM**  \*HLL- struktury formalne jezykow, ASM – brak struktur formalnych  \*HLL – przenośne na rozne platformy, ASM – trudno zrealizować przenośność  \*HLL – w sterownikach urzadzen zlozone techniki programowania, ASM – bezposredni i prosty dostęp do hardwareu  **3. Co to są dyrektywy asemblera?**  Są to rozkazy dla asemblera. Wykonwane przez asembler lecz nie tlumaczone na ML (Język maszynowy). Mogą wpływać na wynikowy kod, tablice symboli, pliki listingu, wewnetrzne parametry asemblera.  **4. Podstawowa roznica pomiedzy makroinstukcją a wywołaniem procedury.**  Markoinstrukcja jest wklejana w miejscu wywołania zaś po wywołaniu procedury następuje skok do mniejsca w którym ją zadeklarowano ( wykorzystywany jest stos do wrzucania wartości powrotu itp.).  **5. Przekazywanie parametrów przez nazwe**  Przekazując parametr przez nazwę podajemy wskaźnik do funkcji obliczającej adres zmiennej.  **6. Wyjaśnij na czym polega zasada pre-processingu przy tłumaczeniu na kod maszynowy.**  Preprocessing polega na tym, że kod źródłowy analizowany jest pod kątem występujących w nim dyrektyw preprocesora. Dyrektywy te są po prostu rozkazami, które rozpoznaje i wykonuje tylko procesor.  **7. Wyjaśnij co to jest parametr wejściowy**  Parametr wejściowy przekazywany do procedury to taki parametr który nie jest modyfikowany przez procedurę, lub jego modyfikacja nie jest widoczna poza procedurą.  **8. Jakie flagi wykorzystuje się przy sprawdzaniu warunku a>=b ?**  Flaga zero ZF i flaga przeniesienia CF  Dla liczb ze znakiem są to: flaga znaku SF i przepełnienia OF  **9. W jakim przypadku nie możemy wykorzystać przekazywania parametrów przez rejestry**  Gdy mamy dużą liczbę parametrów do przekazania, bądź rejestry z których chcemy skorzystać są już w użyciu. Należy wykorzystać wtedy stos.  **10. Co to jest makroinstrukcja?**  Markoinstrukcja jest to symbol, któremu przypisano fragment tekstru programu ( w definicji). Gdziekolwiek w pliku źródłowym znajdzie się symbol zdefiniowanego wcześniej makra assembler wpisze w jego miejsce w pliku źródłowym tekst przypisany do makra.  **11. Implementracja przełącznika CASE**  Krok 1 – utworzenie tablicy zawierającej wyróżniku i przesunięcia do procedur.  Krok 2 – użycie pętli do przeszukania tablicy oraz wywołanie procedury, której przesunięcie jest wpisane w znalezionej pozycji.  **12. Krótko opisać sposób pobierania parametrów z pamięci programu.**  Parametry możemy przekazywać przez:  \*wartość – procedura nie zmienia wartosci parametrow  \*odwołanie – wskaźnik do zmiennej ( adres )  \*zwróconą wartość – przekazywany wskaznik do zmiennnej, wartosc zmiennej kopiowana do zmiennej wewnętrznej, wyjscie – wartosc wyniku  \*rezultat – przekazywanie wyłącznie parametru wyjsciwego \*nazwę – wskaźnik do funkcji obliczającej adres zmiennej  **13. Na czym polega asemblacja warunkowa**  Asemblacja warunkowa jest to generacja innego kodu w zależności od „otoczenia”. Pewne części kodu są włączane do asemblacji po zaistnieniu jakiegoś określonego warunku.  **14. Wyjaśnij zasadę programowania proceduralnego**  Z grubsza idea programowania proceduralnego prezentuje się tak:  1.Zaprojektuj program  2.Podziel na .mniejsze zadania  3.Użyj struktury hierarchicznej opartej na procedurach  4.Poszczególne procedury testuj oddzielnie  **15. Korzyści z przkazywania parametrów przez rejestr**  Przekazywanie parametrów przez rejestr jest najszybszą metodą przekazywania parametrów.  **16. Co powinna zawierać pętla UNTIL**  Dyrektywę .REPEAT , .UNTIL a za nią warunek pętli.  **35. Jak wykonuje się mnożenie liczb zmiennoprzecinkowych?**  Wykonuje się poprzez mnożenie mantys i dodawanie wykładników obu liczb. Po wykonaniu mnożenia – normalizacja ( przesuwanie mantysy w lewo tak, aby usunąć nieznaczące zera – obcięcie najmłodszych bitów wyniku ). W rezultacie normalizacji może nastąpić przepełnienie ( ujemne ) wykładnika.  **36. Podaj wady i zalety stosowania makroinstrukcji**  **Wady**: Jest to mechanizm statyczny -ma określoną postać na etapie generacjiprogramu źródłowego, Marka wplywają na długość programu,  **Zalety**: Redukcja błędów powodowanych przez programistę, umożliwia zdefiniowanieczęsto używanych sekwencji instrukcji, skraca czas przygotowania programu źródłowego.  **37. Co zapewnia lokalność zmiennych w makroinstrukcjach?**  Markoprocesor nadaje unikalne nazwy zmiennym w czasie wstawiania tekstu makrodefinicji do programu. Ten właśnie mechanizm zapewnia lokalność zmiennych  **38. Podaj główne kroki programowej implementacji mnożenia argumentów w kodzie U2**  Mnożnik umieszczony w rejestrze – Q, mnożna w M, istnieje jeszcze 1-bitowy rejestr Q-1 na prawo (logicznie) od Qo. Wyniki mnożenia w rejestrach A i Q. A i Q zerowane. 2. Mnożnik przeglądany bit po bicie, badany też jest bit po jego prawej stronie. Jeśli: oba równe to bity w A, Q ,Q-1 przesuwane arytmetycznie w prawo o jeden. różnią się to zależnie czy mają wartości kolejno 01 czy 10 następuje A+M lub A-M a następnie przesunięcie arytmetyczne w prawo o jeden Dopóki licznik bitów > 0 powtarzaj.  **39. Podaj wybraną zasadę konwersji liczby ze znakiem na przeciwną**  W przypadku konwersji liczby typu rzeczywistego na całkowitą, obetniemy część ułamkową liczby rzeczywistej, w przypadku konwersji char -> int, zamiast znaku będziemy mieli do dyspozycji liczbę (czyli kod ASCII), pod jaką kryje się dany znak itd..   1. **40.Opisz zasadę dzielenia liczb w zapisie zmiennoprzecinkowym**    1. Jeśli dzielnik = 0 zgłaszany jest błąd lub wynik ustalany jako nieskończoność. ( w zależności od implementacji )    2. Jeśli dzielna = 0 wynik =0    3. Jeśli żadne z powyższych to wykładnik dzielnika jest odejmowany od wykładnika dzielnej.    4. Dodawane przesunięcie    5. Jeśli nastąpiło przepełnienie lub niedomiar wykładnika – zgłoś.    6. Dzielenie mantys. Normalizacja 2. **41.Opisz zasadę dodawania liczb w zapisie zmiennoprzecinkowym.**    1. Aby dodać dwie liczby zmiennoprzecinkowe należy przeprowadzić normalizacje.    2. Normalizacja – taka zmiana zapisu mantysy jednej z liczb aby jej wykladnik był rowny wykladnikowi drugiej    3. Liczbę o mniejszym wykladniku dopasowuje się do liczby o większym. Dodawanie liczb realizuje się poprzez dodanie znormalizowanych mantys obu liczb. 3. **44. Co powinna zawierać markodefinicja?**    1. Nazwę Makra ( etykieta )    2. Dyrektywę „MAKRO”   Dyrektywę „ENDM**C:\Users\Vania\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Przechwytywanie.jpg**  **44.Podaj opisowo co powinna zawierać implementacja pętli REPEAT i jaka powinna być sekwencja działań.**  Powinna posiadać etykietę na początku pętli oraz instrukcję skoku do etykiety pod określonym, wymaganym w danej sytuacji warunkiem.   * 1. Wykonuj instrukcję wewnątrz pętli   2. Po napotkaniu instrukcji skoku:      1. jeśli warunek spelniony skok do etykiety      2. jeśli warunek niespełniony przejdź do następnej instrukcji  1. **45.Co musi wynikać z deklaracji zmiennych lokalnych zadeklarowanych dyrektywą taką jak LOCAL?**   Należy określić symbol i jego typ. Symbol może być zmienną prostą lub tablicową.  **46.Jak markoasembler tłumacząc procedurę zadeklarowaną dyrektywą taką jak PROC skonstruuje dostęp do parametrów przekazywanych do procedury?**  Makroasemb Makroasembler implementuje przekazanie parametrów poprzez stos: \*W programie generuje kod zapisania parametrów na stos  \*W procedurze generuje kod realizujący dostęp do parametrów na stosie (obsługa wskaźnika ramki); symbole parametrów zastępuje wyrażeniami adresowymi ( przesunięciami w obrębie ramki stosu )   1. **47.Kiedy można używać przekazywania parametrów w kodzie programów?**   Gdy jesteśmy pewni, że parametr zawsze będzie miał tą samą wartość.  **48. Kiedy można parametry do procedury przekazywać przez rejestr?**  Przy odwoływaniu pośrednim i bezpośrednim. Gdy kod wywołania procedury uwzględnia zapisanie potrzebnych danych z rejestru na stos. | **17. Jak wytłumaczyć przekazywanie parametru przez odwołanie**  Gdy przkeazujemy procedurze parametr przez odwołanie przekazujemy do procedury jej adres ( Na stos jest odkładany adres parametru). Może być on następnie bezpośrednio używany i modyfikowany przez procedurę.  18.**Dlaczego przenoszenie oprogramowania systemowego jest trudne lub wręcz niemożliwe?**  Ponieważ język asemblera jest związany z konkretną rodziną procesorów.  **19.Czym się różni parametr wejściowy od wejściowo-wyjściowego?**  Procedura nie zmienia wartości parametru wejściowego, albo nie jest to widoczne poza procedurą. Parametr wejściowo-wyjściowy jest wskaźnikiem do zmiennej (adres) zawierającej wartość która jest używana i modyfikowana przez procedurę.  **20. Przekazywanie danych przez nazwę.**  Przekazany zostaje wskaźnik do funkcji obliczającej adres zmiennej  **21. Wyjaśnij co to jest parametr wejściowy**  Wywołana procedura nie będzie modyfikować zmiennej odpowiadającej temu parametrowi (ew. modyfikacja nie będzie widzialna poza procedurą.   1. **Na co należy zwrócić uwagę przy wykonywaniu złożonej operacji warunkowej, np. a>b AND c>d**   Należy kod instrukcji skonstrułować w taki sposób aby wykonywanych było jak najmniej porównań. Kod taki będzie krótszy i będzie szbyciej się wykonywał co wpłynie na efektywność programu.   1. **W jakim przypadku uzasadnione jest wykorzystywanie przekazania parametru przez blok.**   W przypadku dużej ilości parametrów. Jest to bezpieczniejsze ( nie przeciążamy stosu ) i wygodniejsze ( przy bardzo duzej ilości parametrów ) niż inne formy przekazywania parametrów.  **24Jakie miejsce zajmuje oprogramwanie systemowe w strukturze oprogramowania?**  Realizuje funkcje konieczne dla działania systemu komputerowego(świadczy usłguiprzetowi komputerowemu), w szczegolnosci steruje i koordynuje prace komputera zapewniajac jego funkcjonalnosc, zwłaszcza możliwość wykonywania programow aplikacyjnych, w sklad o.s wchodzi kazde oprogramowanie z którym użytkownik nie ma bezposreniej styczności, pojecie szersze niż systemy operacyjne.  **25. Kiedy lepiej jest zastąpić procedurę makroinstrukcją?**  Kiedy chcemy zaoszczędzić na czasie wykonywania programu wynikowego (brak wielu operacji związanych z przekazaniem parametrów do procedury i rozkazu skoku do procedury = oszczędnosć czasu ) i jednocześnie nasz program nie jest zbyt długi ( użycie makr zwiększa długość programu wynikowego ).   1. **Kiedy uzasadnione jest przekazywanie parametrów przez komórki w pamięci programu?**   Gdy dane podane jako parametry będą później wykorzystywane?   1. **Na czym polega normalizacja wykonywana po mnożeniu liczb w zapisie zmiennoprzecinkowym?**   Jest to przesunięcie mantysy w lewo tak, aby usunąć nieznaczące zera - obcięcie najmłodszych bitów wyniku. W efekcie może nastąpić przepełnienie ujemne wykładnika.   1. **Jak bezpiecznie deklarować zmienne stałe bez względu na to jakiego są typu?**   Dyrektywą EQU ?  **29. W jakich jfjjgjmomentach następuję przepełnienie w dodawniu liczb zmiennoprzecinkowych**  Przy dodawaniu mantys z uwzględnieniem znaków. Może dojśc do przepełnienia mantysy lub wykładnika  **30. Jak wygląda przekazanie parametru w makrze?**  Podczas analizowania programu przez makrogenerator w miejsce odwołań do makropoleceń są wstawiane treści makropoleceń. Jednocześnie jest dokonywana zamiana parametrów formalnych na aktualne przy czym i-ty parametr aktualny odpowiada i-temu parametrowi formalnemu. Jeśli liczba aktualnych parametów jest większa niż formalnych to dodatkowe parametry są ignorowane. W sytuacji przeciwnej makroasembler przyjmuje, że brakujące parametry są pustymi tekstami.  **31. Co to jest normalizacja przy dodawaniu liczb zmiennoprzecinkowych?**  To taka zmiana zapisu mantysy jednej z liczb aby jej wykladnik równy był wykładnikowi drugiej liczby. Liczbę o mniejszym wykładniku dopasowuje się do liczby o wykładniku większym.  **32. Co zyskamy kiedy zamiast wykoniania x/z\*y wykonamy x\*y/z**?  Jeżeli X\*Y nie powoduje przepełnienia, dokładniejszy wynik uzyskamy implementując: W:=X\*Y/Z.   1. **Jak odbywa się przekazanie parametrów do rejestrów w procedurze?** 2. **Opisać implementację realizacji bloku warunkowego X przy warunku (a>b) or (b>c)**   cmp a,b ; A>B?  ja L1 ; tak  cmp b,c ; nie: B>C?  jbe next ; nie: omiń  L1: polecenie ; wykonaj polecenie  next: |