

ЛР 5 (с повторами для каждого варианта)

1. Какие разновидности команды XLAT Вы знаете, и чем они отличаются?
2. Как организовать ожидание ввода символа в Ассемблере?
3. Какие системы счисления Вы знаете?
4. В чем отличие символьного и шестнадцатеричного представления в ОП?
5. Можно ли в процедуру Ассемблера передать параметры при ее вызове и как?
6. Что такое стек и для чего он нужен?
7. В каких командах Ассемблера неявно используется стек?
8. Поясните работу команды CMP?
9. Поясните работу команды условного перехода - JE/JNE?
10. Какие регистры неявно используются в команде LOOP?
11. Поясните работу команды LOOP?
12. Как организовать вложенный цикл в Ассемблере?
13. Как по листингу определить размер программы в оперативной памяти (ОП)?
14. Как по листингу определить длину фрагмента программы, указанного преподавателем?
15. Чем отличаются *.COM и *.EXE исполнимые файлы, построенные Ассемблером?
16. Как можно в программе ассемблера организовать ветвление?
17. Поясните команды условного и безусловного перехода.
18. Какие команды сравнения Вы знаете?
19. Какие системы счисления Вы знаете?
20. Поясните работу команды CMP?
21. Поясните работу команды LOOP?
22. Как организовать вложенный цикл в Ассемблере?
23. Доп. Чем отличаются *.COM и *.EXE исполнимые файлы, построенные Ассемблером?
24. Как организовать вывод символа на экран в Ассемблере?
25. Какие команды условного перехода Вы знаете?

ЛР 6 (с повторами для каждого варианта)

1. Что такое PSP? Его основные данные.
2. Для каких целей PSP может использоваться в программах?
3. Какая информация хранится в PSP и как с ней работать?
4. Как можно узнать число байт введенных параметров в командной строке?
5. Доп. Как можно определить число введенных параметров?
6. Где можно получить информацию о структуре PSP?
7. Как выполнить доступ к PSP в *.COM файле?
8. Как выполнить доступ к PSP в *.EXE файле?
9. Поясните работу команд CALL и RET с учетом изменения и использования стека?
10. Доп. Как можно использовать в программе счетчик команд – оперативной памяти (\$)?
11. Чем отличаются *.COM и *.EXE исполнимые файлы, построенные Ассемблером?
12. Какие классы прерываний Вы знаете?
13. Каким символом разделяются отдельные параметры при запуске программ?
14. Какое максимальное число параметров можно задать в командной строке (вычислите)?
15. Как Вы определили число параметров в своей программе?
16. Что такое БНФ и как ее используют программисты?
17. Доп. Какие данные из PSP Вы можете использовать в своих программах?
18. Какие виды прерываний Вы знаете?
19. В чем главное отличие программных и аппаратных прерываний?
20. Как сделать прерывания недоступными?
21. Поясните работу команд INT и IRET с учетом изменения и использования стека?
22. Как выполнить доступ к PSP в *.COM файле?
23. Как выполнить доступ к PSP в *.EXE файле?
24. Где можно получить информацию о структуре PSP?

25. Как сделать прерывания снова доступными?

ЛР 7 (с повторами для каждого варианта)

1. Поясните назначение регистров: CS, DS, SS и ES.
2. Поясните назначение регистров: IP, BP, SP, DI и SI.
3. Поясните назначение регистров: AX, BX, CX и DX.
4. Что такое короткий адрес (NEAR)?
5. Что такое длинный адрес (FAR)?
6. Как в отладчике посмотреть содержимое стека?
7. Доп. Как в отладчике посмотреть содержимое оперативной памяти?
8. Как получить адрес и местоположение PSP?
9. Какие регистры МП управления вы знаете? Их назначение?
10. Для чего нужны сегментные регистры? Перечислите их.
11. Как в отладчике выполнить анализ выполняемой команды? Что для этого нужно сделать?
12. Как запустить компилятор без формирования отладочной информации?
13. Как запустить TLINK (или LINK) без формирования отладочной информации?
14. Для чего нужна утилита make.exe? Как ее использовать?
15. Какие шаги нужно выполнить для формирования исполнимого модуля на Ассемблере?
16. Доп. Как при редактировании связей подключить объектные модули (*.obj) из библиотеки
17. Как определить в программе на языке Ассемблера адрес выполняемой команды?
18. Поясните назначение регистров: CS, DS, SS и ES.
19. Поясните назначение регистров: IP, BP, SP, DI и SI.
20. Поясните назначение регистров: AX, BX, CX и DX.
21. Доп. Как запустить TASM (или MASM) без формирования отладочной информации?
22. Как запустить TASM (или MASM) с формированием отладочной информации?
23. Доп. Как запустить TLINK (или LINK) без формирования отладочной информации?
24. Как запустить TLINK (или LINK) с формированием отладочной информации?
25. Для чего в программе и как заноситься регистр DS?

ЛР 8 (с повторами для каждого варианта)

1. Какую функцию выполняет команда LEA?
2. Какую функцию выполняет команда LES?
3. Какую функцию выполняет команда LDS?
4. Что такое ближний (NEAR) и дальний (FAR) адрес?
5. Как задается ближний (NEAR) и дальний (FAR) адрес?
6. Поясните назначение регистров IP и EIP? В чем их отличие?
7. Поясните назначение регистров AX и EAX? В чем их отличие?
8. Поясните назначение регистров BX и EBX? В чем их отличие?
9. Поясните назначение регистров CX и ECX? В чем их отличие?
10. Поясните назначение регистров DX и EDX? В чем их отличие?
11. Какие способы вы можете назвать для очистки экрана (минимум 2)?
12. Как организовать в программе Ассемблера вложенный цикл (с помощью 2-х команд LOOP)?
13. Как при выводе дампа памяти проверить выход за границу сегмента?
14. Поясните назначение регистров IP и EIP? В чем их отличие?
15. Поясните назначение регистров AX и EAX? В чем их отличие?
16. Поясните назначение регистров BX и EBX? В чем их отличие?
17. Поясните назначение регистров CX и ECX? В чем их отличие?
18. Поясните назначение регистров DX и EDX? В чем их отличие?
19. Какую функцию выполняет команда LEA?
20. Какую функцию выполняет команда LES?
21. Какую функцию выполняет команда LDS?

- 22. Доп. В чем отличие между 16-ти и 32-ти разрядными приложениями?
- 23. Доп. Какие компиляторы нужно использовать для 16-ти и 32-ти разрядных приложений?
- 24. Доп. Что необходимо сделать для подключения собственного обработчика прерывания?
- 25. Доп. Какими способами можно проверить наличие в памяти собственного обработчика прерывания?

На вопросы, помеченные сочетанием “Доп.” (дополнительные), можно не отвечать!!!!