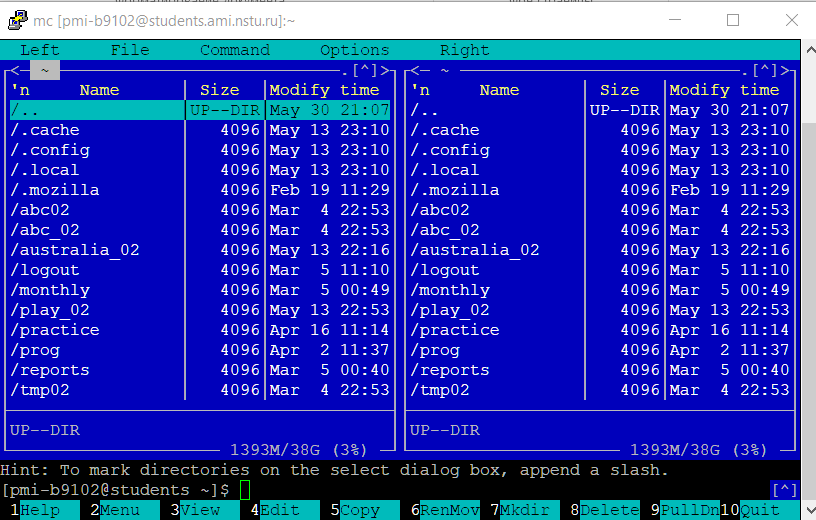
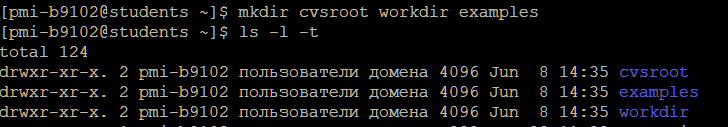
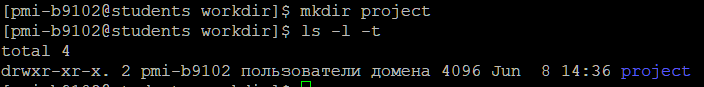
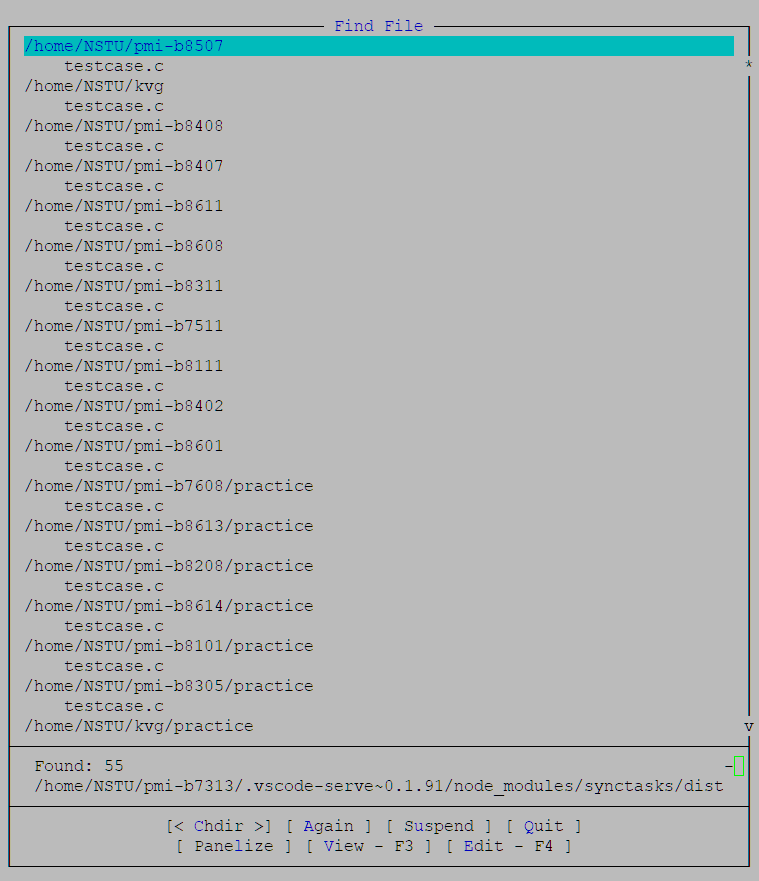
|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
|  | | |
| Лабораторная работа № 5 | | |
| по дисциплине «Операционные системы» | | |
| Инструментальные средства разработки программ | | |
|  | | |
|  | Группа | ПМ-91 |
| Бригада |  |
| 2 |
| Студенты | Затолоцкая Юлия,  Константинова анастасия |
| Преподаватель | Кобылянский Валерий Григорьевич |
| Дата | 09.06.21 |
|  | | |
|  | | |

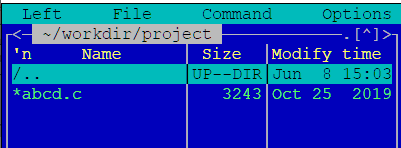
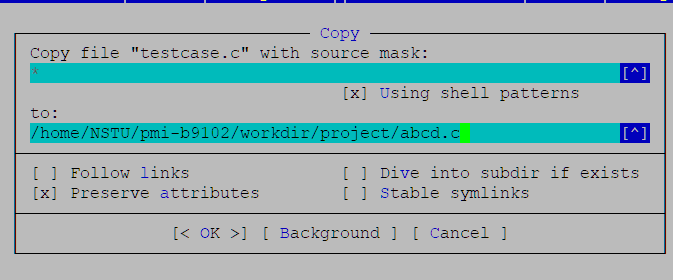
**1. Цель работы**

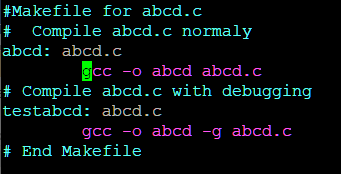
Целью работы является изучение основных этапов разработки и отладки приложений в ОС Linux, а также приобретение практических навыков по использованию инструментальных средств фонда свободного программного обеспечения при компиляции исходного кода, сборке, отладке и тестировании программ, написанных на языке Си.

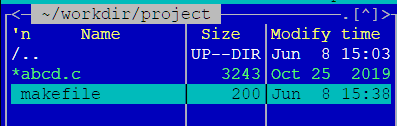
**2. Ход работы**

1. Запустите файловый менеджер **mc**
2. В домашнем каталоге создайте каталоги **cvsroot,** **workdir** и **examples,** в каталоге **workdir** создайте подкаталог **project**
3. Выполните поиск во внешней памяти сервера файла **testcase.c**



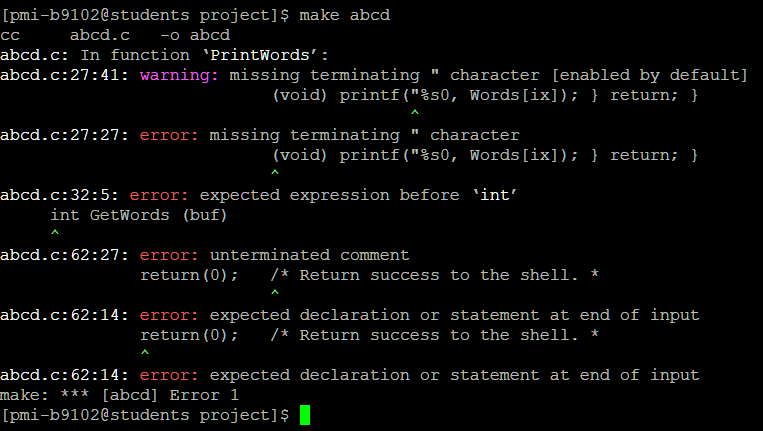
1. Скопируйте файл **testcase.c** в каталог **project** под именем **abcd.c** и занесите в отчет исходный текст программы, предварительно сравнив его с текстом, приведенныв в п.2.6.
2.   С помощью редактора **vi** создайте в каталоге **project** make-файл согласно п. 2.3.



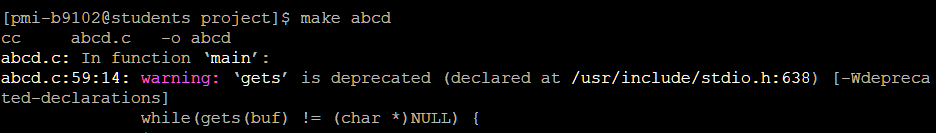


1. Выполните компиляцию программы **abcd.c.** с помощью make-файла, используя правило **abcd**. При необходимости исправьте синтаксические ошибки с помощью редактора **vi,** информацию по ошибкам и их устранению занесите в отчет (номер строки, значение строки до устранения и после устранения ошибки, пояснения).



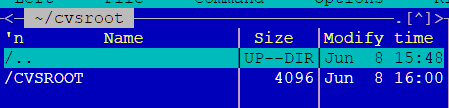
Не было tab перед командой, добавили.

|  |  |  |
| --- | --- | --- |
| Номер строки | Код | Исправленный вариант |
| 27 | (void)printf("%s0, Words[ix]); } return; } | (void)printf("%s", Words[ix]); } return; } |
| 62 | return(0); /\* Return success to the shell. \* | return(0); /\* Return success to the shell. \*/} |

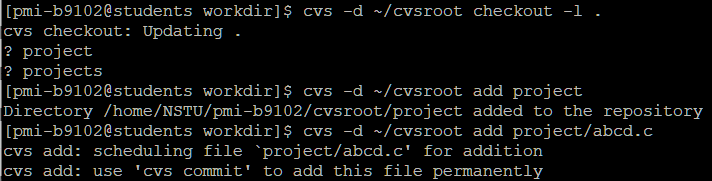
В строке 27 заменили 0 на «"». В строке 62 добавили «/» для завершения конструкции комментария /\*…\*/, добавили в конец «}», для завершения тела функции main.

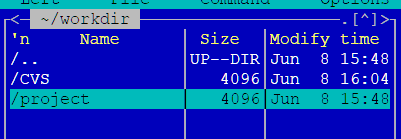
1. В каталоге **cvsroot** создайте репозиторий CVS.

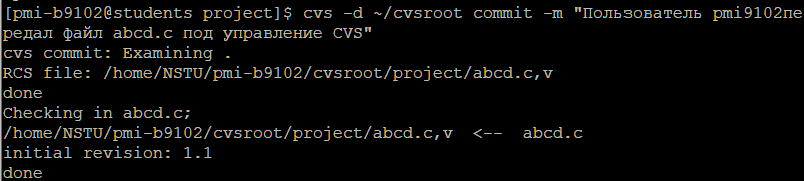




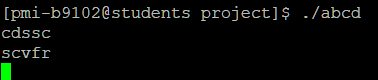
1. Передайте каталог **project** и файл **abcd.c** в репозиторий. При выполнении команды **commit** с помощью редактора **vi** введите комментарий, например: “Пользователь pmi7501передал файл abcd.c под управление CVS”.





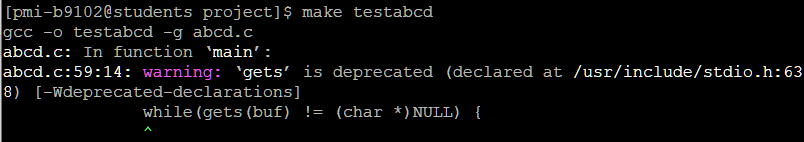


1. Запустите исполняемый файл abcd и поясните результат запуска.

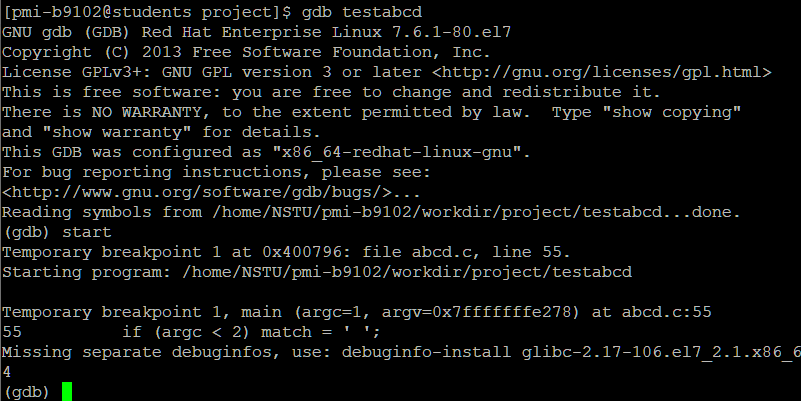


После запуска программа начинает бесконечно считывать текст, но ничего не выводить. Вероятнее всего это происходит из-за семантических ошибок в программе.

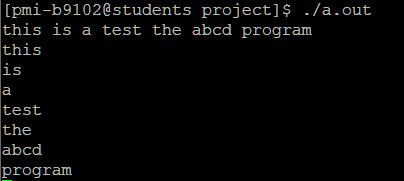
1. Перекомпилируйте программу с помощью правила **testabcd** make-файла.

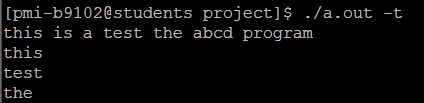


1. С помощью отладчика **gdb** выполните поиск и устранение семантических ошибок в программе **abcd**. Каждое исправление в программе должно сопровождаться записью в репозиторий новой версии с комментарием, поясняющим на русском языке сущность исправлений (например, номер строки программы **abcd.c** и причина исправления). После устранения всех ошибок занесите в отчет результаты тестирования программы в двух вариантах запуска – без параметра и с параметром.

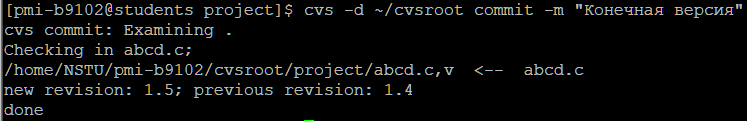
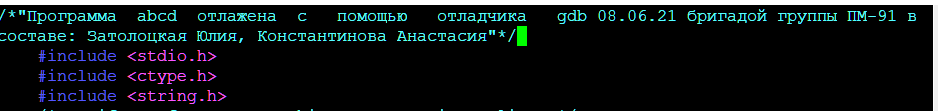


|  |  |
| --- | --- |
| Номер строки | Исправление |
| 55 | Было: match = ‘ ‘;  Стало: match = ‘\0‘; |
| 27 | Было: (void) printf(“%s”, Words[ix]);  Стало: (void) printf(“%s\n”, Words[ix]); |
| 25 | Было: while ((\*cp) && (\*cp++ != match));  Стало: while ((\*cp) && (\*cp != match)) cp++; |

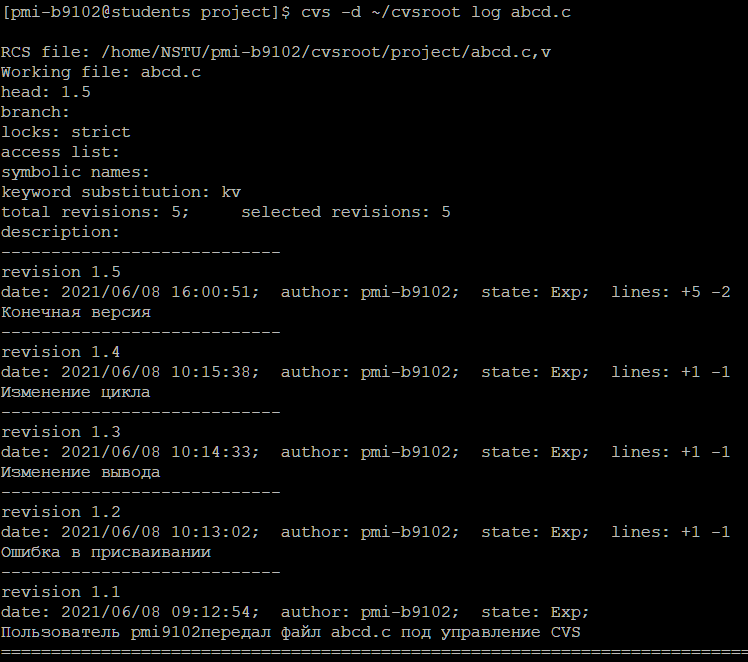




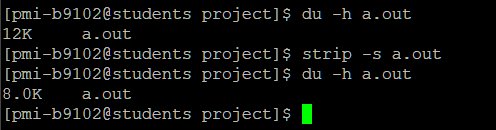
1. После получения корректных результатов выполнения программы **abcd** с помощью редактора **vi** в начало отлаженной программы введите комментарий: "Программа abcd отлажена с помощью отладчика gdb дд.мм.гг. бригадой группы ПМ-ХХ в составе: ФИО1, ФИО2..." и сохраните в репозиторий финальную версию программы.



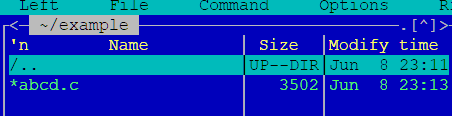
1. Выведите список изменений файла **abcd.c**, выполненных в ходе отладки программы, занесите список в отчет.



1. Определите размер исполняемого модуля отлаженной программы. Удалите всю отладочную информацию и снова определите размер исполняемого модуля, сравните с предыдущим результатом, результат сравнения занесите в отчет.



1. Извлеките из репозитория полностью отлаженную программу **abcd** и скопируйте её в каталог **example**, заменив в нем предыдущую версию программы. Все дальнейшие действия будут выполняться в этом каталоге.



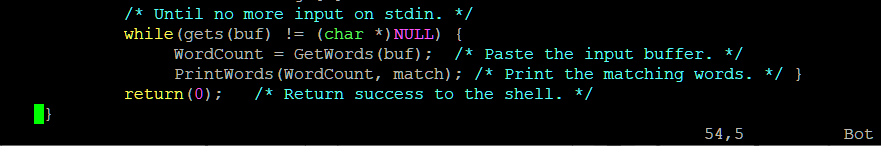
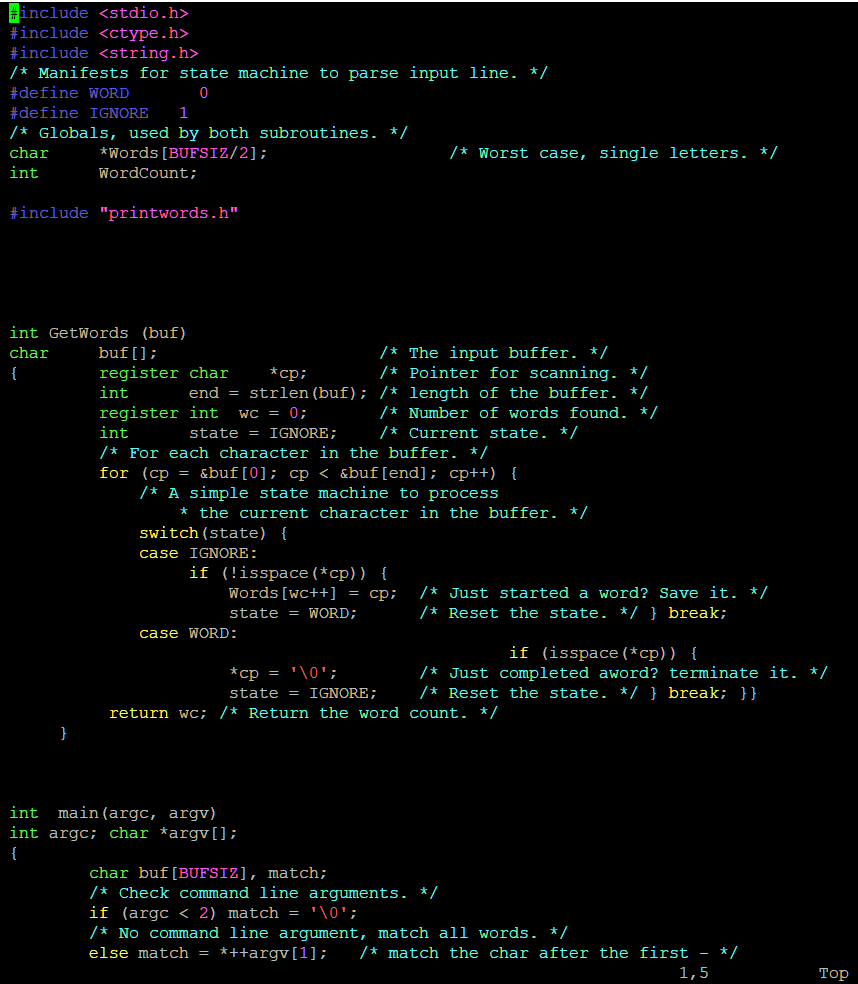
1. Выполните разбиение полностью отлаженной программы **abcd** на функции в соответствии с номером бригады из таблицы 26. Обратите внимание на тип функции (внутренняя или внешняя), тип файла (.c, .h или .o) и тип модуля (исходный или объектный). Занесите в отчет измененный текст программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № бригады | main() | | printwords() | | getwords() | |
|  | тип | файл | тип | файл | тип | файл |
| 2 | внутренний, исходный | abcd2.c | внешний, исходный | printwords.h | внутренний, исходный | abcd2.c |

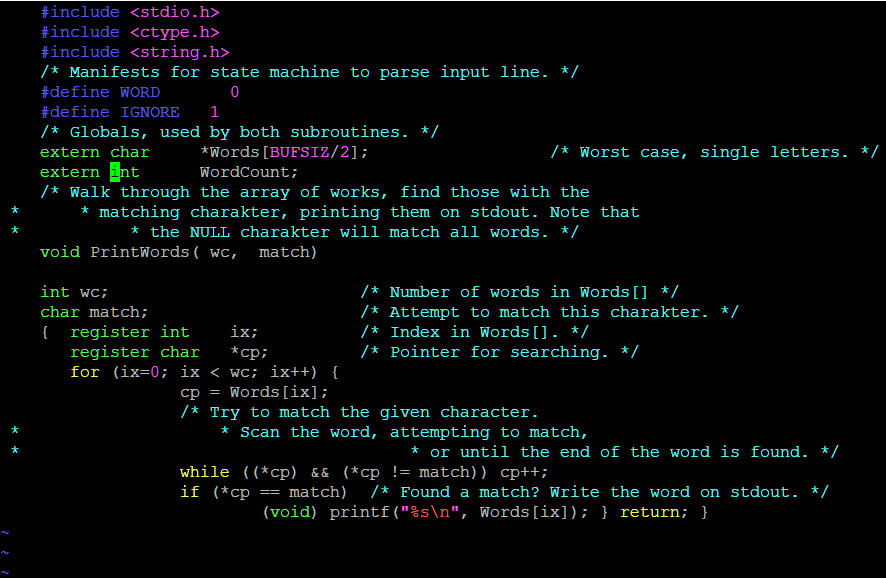
Создадим файлы:



Файл abcd2.c:

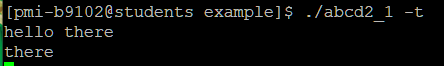


Файл printwords.h:

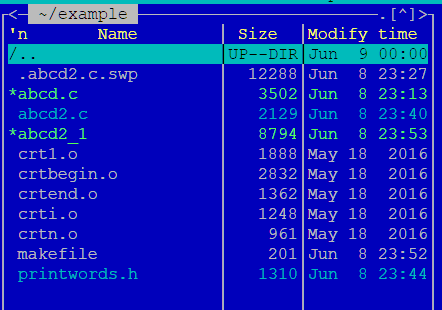
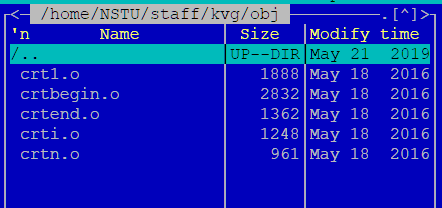


1. Выполните сборку программы в соответствии вариантом задания, используя неявный вызов компоновщика и задав имя исполняемого файла **adcd2\_1;** проверьте корректность работы программы и занесите в отчет результаты ее тестирования.

gcc abcd2.c printwords.h –o abcd2\_1



1. Выполните поиск во внешней памяти сервера каталога **obj**, скопируйте из него все файлы в каталог **example**, поясните в отчете назначение скопированных файлов.



|  |  |
| --- | --- |
| crt1.o | содержит код, инициализирующий среду исполнения языка C и вызывающий определенную пользователем функцию main |
| crtbegin.o, crtend.o | обрабатывают глобальные конструкторы и деструкторы языка С++ |
| crti.o | содержит пролог функции \_init, помещаемый в начало секций .init |
| crtn.o | содержит эпилог функции \_init, помещаемый в конец секций .init |

1. Выполните сборку программы в соответствии вариантом задания, используя явный вызов компоновщика. Результатом сборки должны быть исполняемый файл **abcd2\_2** и карта памяти **abcd2\_map;** проверьте корректность работы программы и занесите в отчет результаты ее тестирования.

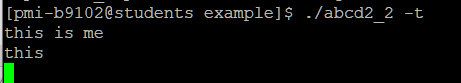
Создаем объектные модули:



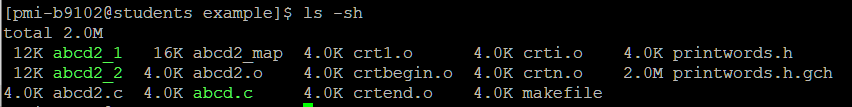
Выполняем явный вызов компоновщика:

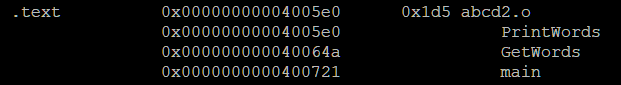






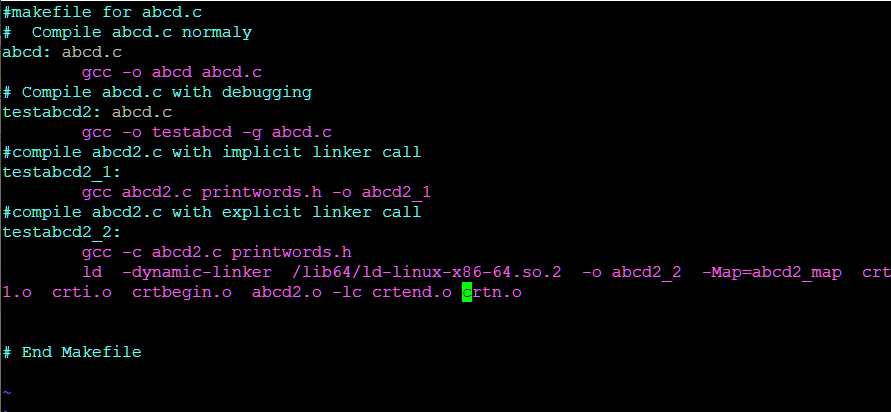
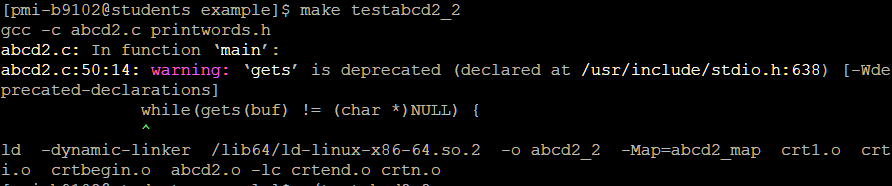
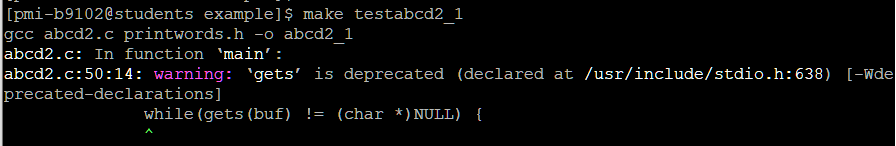
1. Из карты памяти **abcd2\_map** определите размеры машинного кода модулей **abcd2.o,** **printwords.o** и **getwords.o,** сравните их с размерами исходного и объектного кода этих модулей (файлы типа **.с** и **.о)**. Результат представьте в виде таблицы 27, все данные должны быть подтверждены скриншотами.





|  |  |  |  |
| --- | --- | --- | --- |
| Имя модуля (функции) | Исходный, байт | Объектный, байт | Машинный код, байт |
| printwords | 4000 | - | 0x400721-0x4005e0 |
| getwords | 4000 | - | 0x400721-0x4005e0-0x40064a |
| abcd2 | 4000 | 4000 | 0x1d5 |

1. Добавьте в make-файл, разработанный при выполнении п.5, два новых правила, реализующие п. 17 и 19 задания. Проверьте корректность его работы.

**3. Выводы**

Контрольные вопросы проработаны.