

Задание №0 в рамках вычислительного практикума

Настройка окружения

Кострицкий А. С., Ломовской И. В.

Москва — 2022 — TS2202081853

Содержание

1	Цель работы	1
2	Задание	1
3	Список пакетов	1
4	Формат защиты	2

1 Цель работы

Целью данной работы является настройка необходимого окружения для работы дома.

2 Задание

Требуется:

1. Настроить домашнее окружение, установив определённый далее набор пакетов.
2. Проверить работоспособность окружения, ответив на представленные далее вопросы.

3 Список пакетов

Список ниже специально представлен в неудобном для копирования в терминал виде, чтобы Вам пришлось перед установкой каждого хотя бы на секунду задуматься.

1. Интерпретатор Питона с обвязкой и менеджером пакетов:

```
python3 pip3
```

2. Пакеты, используемые при выполнении лабораторных работ по первой части курса «Программирование на Си»:

```
git gcc valgrind clang clang-tools gdb
```

3. Среда разработки, установленная в лабораторных залах:

qtcreator

Допускается самостоятельная работа в любой альтернативной среде программирования (VS Code, CLion, Geany, etc.), позволяющей:

- (a) вызывать свой, а не встроенный в среду изначально, компилятор;
- (b) настраивать этапы сборки самостоятельно;
- (c) проводить трассировку приложения.

Обращаем Ваше внимание, что экзамен проводится в лабораторных залах, где настроен только Qt Creator.

4. Пакеты для выполнения лабораторных работ по второй части курса «Программирование на Си»:

make

5. Пакеты для выполнения заданий в рамках практикума:

diffutils gnuplot shellcheck

6. Для рисования диаграмм в нотации IDEF0 в курсе «Основы программной инженерии» используется программа Ramus, опубликованная на [GitHub](#) в формате jar. Для её работы требуется пакет:

java

Лабораторные работы по курсу «Основы программной инженерии» проводятся в залах на машинах с ОС Windows, но все используемые пакеты бесплатны и доступны на Linux – **по договорённости с проводящим занятия преподавателем** можно выполнять лабораторные и на Linux.

7. Пакет для автоматического документирования кода в курсе «Основы программной инженерии»:

doxygen

8. Полезно иметь:

nano, htop, ffmpeg

4 Формат защиты

Студентом заполняется одна страница документа в формате docx (odt), в которой должны быть ответы на следующие вопросы (разрешается в виде таблицы), сами вопросы не копируются. Страница документа и её pdf-версия загружаются как ответ на задание в кафедральный мул.

Если при ответе на какой-либо вопрос Вы столкнулись с проблемами, опишите их и способ решения.

1. Каковы характеристики Вашего компьютера: процессор, объём доступной памяти? Под управлением какой ОС? Если есть желание, можете указать характеристики дополнительного оборудования.
2. Если Вы работаете на виртуальной машине, каковы её характеристики?
3. Какова версия установленного у Вас компилятора gcc? Что Вы видите в ответ на вызов `gcc -v`?
4. Какова версия установленного у Вас компилятора clang? Что Вы видите в ответ на вызов `clang -v`?
5. Можете ли у Вас с помощью компилятора gcc оттранслировать файл исходного кода следующего содержания:

```
include <stdio.h>
int main(void)
(
return 0;
)
```

6. Скопируйте файл `.bashrc` в `.bashrc.old`. Что Вы видите в ответ на вызов

```
diff .bashrc .bashrc.old
```

Образец отчёта:

Задание №0

Проблемы, возникшие при выполнении задания

Проблем не возникло.

Характеристики ПК и ОС

2Gb RAM, 1.1GHz arm64, 16Gb SSD.

uname -a

```
pi@1405-PI:~ $ uname -a
Linux 1405-PI 5.10.95-v8+ #1519
SMP PREEMPT Tue Feb 1 13:19:30 GMT 2022 aarch64 GNU/Linux
```

gcc -v

```
pi@1405-PI:~ $ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/aarch64-linux-gnu/10/lto-wrapper
Target: aarch64-linux-gnu
Configured with:
../src/configure -v --with-pkgversion='Debian 10.2.1-6'
--with-bugurl=file:///usr/share/doc/gcc-10/README.Bugs --enable-languages=c,ada,
c++,go,d,fortran,objc,obj-c++,m2 --prefix=/usr --with-gcc-major-version-only --program-s
--program-prefix=aarch64-linux-gnu- --enable-shared --enable-linker-build-id
--libexecdir=/usr/lib --without-included-gettext --enable-threads=posix
--libdir=/usr/lib --enable-nls --enable-bootstrap --enable-clocale=gnu
--enable-libstdcxx-debug --enable-libstdcxx-time=yes
--with-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-libquadmath
--disable-libquadmath-support --enable-plugin --enable-default-pie --with-system-zlib --
--with-target-system-zlib=auto --enable-objc-gc=auto --enable-multiarch --enable-fix-cor
--build=aarch64-linux-gnu --host=aarch64-linux-gnu --target=aarch64-linux-gnu --with-bui
--enable-link-mutex
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 10.2.1 20210110 (Debian 10.2.1-6)
```

clang -v

```
pi@1405-PI:~ $ clang -v
Debian clang version 11.0.1-2
Target: aarch64-unknown-linux-gnu
Thread model: posix
InstalledDir: /usr/bin
Found candidate GCC installation: /usr/bin/../../lib/gcc/aarch64-linux-gnu/10
Found candidate GCC installation: /usr/lib/gcc/aarch64-linux-gnu/10
Selected GCC installation: /usr/bin/../../lib/gcc/aarch64-linux-gnu/10
Candidate multilib: .;@m64
Selected multilib: .;@m64
```