



Homework Assignment #1

PID manager

Professor : Hsung-Pin Chang

TA : Wei-Ming Su



Outline

■ Environment

- ☐ Install Ubuntu on VirtualBox
- ☐ Build your program

■ Background

- ☐ What is the PID ?
- ☐ What a PID manager does ?
- ☐ Bitmap

■ Homework Assignment #1

■ Reference



Outline

■ Environment

- ☐ Install Ubuntu on VirtualBox
- ☐ Build your program

■ Background

- ☐ What is PID ?
- ☐ What PID manager do ?
- ☐ Bitmap

■ Homework Assignment #1

■ Reference

Install Ubuntu on Virtual

■ Download VirtualBox

□ <https://www.virtualbox.org/>



VirtualBox

Welcome to VirtualBox.org!

VirtualBox is a powerful x86 and AMD64/Intel64 [virtualization](#) product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. See "[About VirtualBox](#)" for an introduction.

Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of [guest operating systems](#) including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8, Windows 10), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x and 4.x), Solaris and OpenSolaris, OS/2, and OpenBSD.

VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.

[About](#)
[Screenshots](#)
[Downloads](#)
[Documentation](#)
 [End-user docs](#)
 [Technical docs](#)
[Contribute](#)
[Community](#)

Download VirtualBox 6.0

Install VirtualBox and Ubuntu (cont.)

■ Download Ubuntu

- ☐ <http://www.ubuntu-tw.org/modules/tinyd0/>
- ☐ 16.04 LTS
- ☐ 32-bit

下載 Ubuntu

發行版

- ☒ Ubuntu 桌面版本
- ☐ Ubuntu 伺服器版本

不同發行版具備不同的圖形環境與配套軟體。如果您不知道如何選擇，請選擇 Ubuntu 桌面版本。

版本

- ☐ 18.04 LTS (支援至 2023 年 04 月)
- ☒ 16.04 LTS (支援至 2021 年 04 月)
- ☐ 14.04 LTS (支援至 2019 年 04 月)

最新長期支援 (穩定, 五年支援) 版為 18.04 LTS (2018/04 - 2023/04)。16.04 及之前版本的預設桌面為經典 Unity, 18.04 開始為 Gnome 3, 但您仍可手動安裝 Unity 並進行切換。
詳細資訊請參考 [Release End of Life](#)

電腦架構

- ☒ 32 位元版本
- ☐ 64 位元版本

目前一般電腦大多使用 64 位元架構, 故從 18.04 開始, 桌面版也將不再提供 32 位元版本。另外, 目前在下載區無 Mac 版本可下載。

Install VirtualBox and Ubuntu (cont.)

- Install & setup VirtualBox

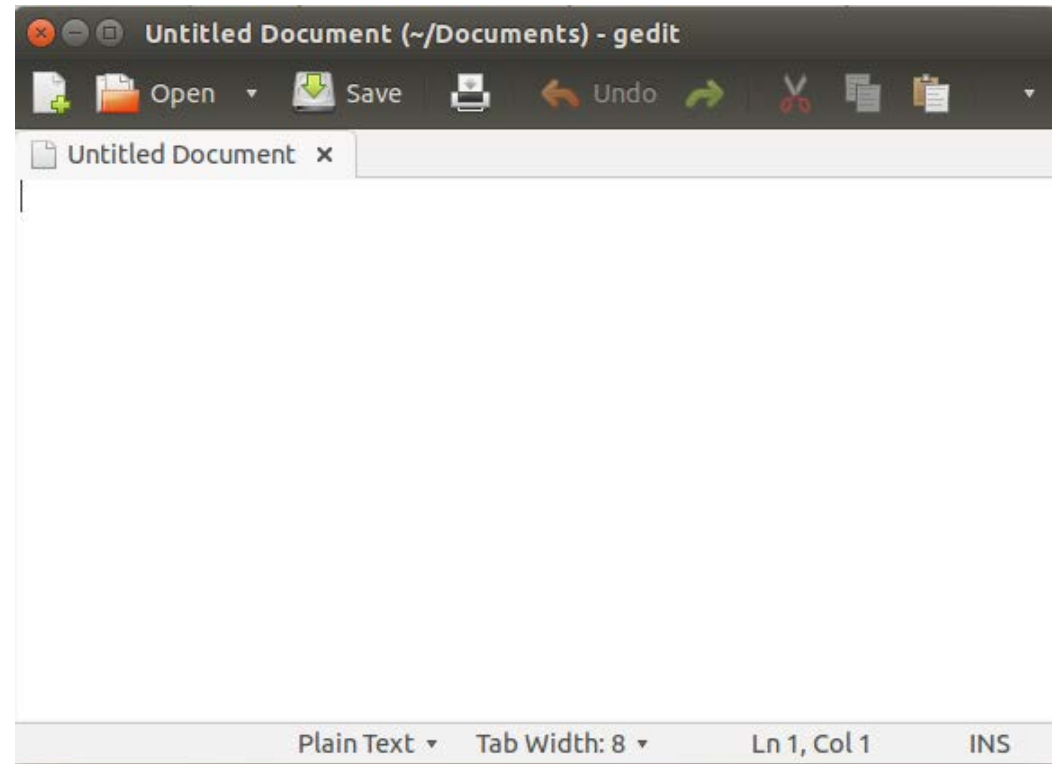
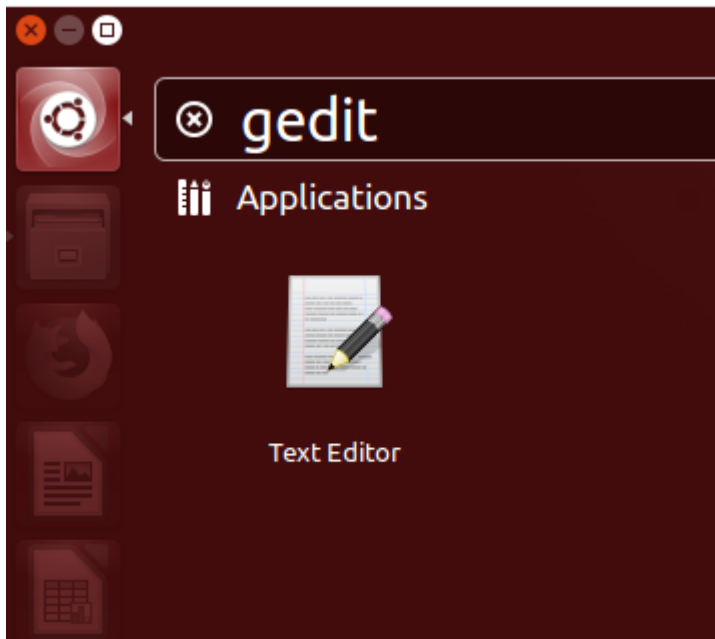
- <https://blog.xuite.net/yh96301/blog/66578586>

- Install Ubuntu

- <https://blog.xuite.net/yh96301/blog/432341564-VirtualBox+5.2%E5%AE%89%E8%A3%9DUbuntu+16.04>

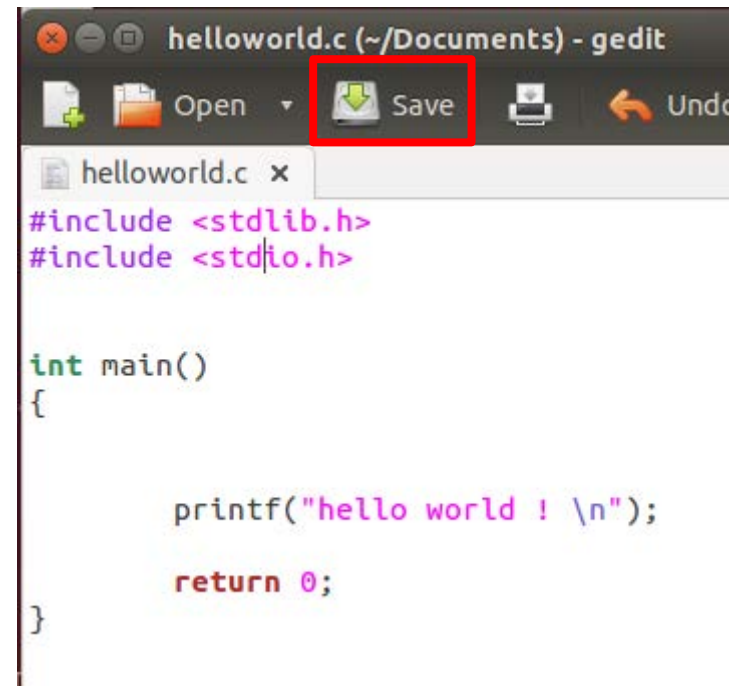
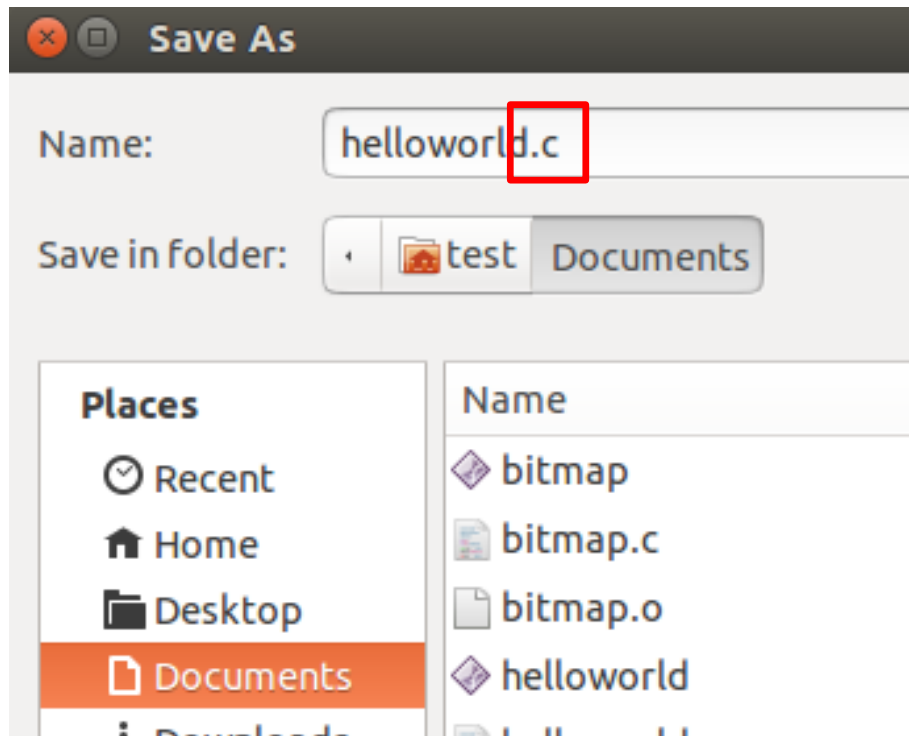
Build your program

- Write your program with **gedit**



Any other editors, either 純文字編輯器 (vi, vim, nano) or 圖形化編輯器 (sublime, notepad, geany) are also available on Ubuntu .

Build your program(cont.)



Build your program

■ Other Editors



VIM - Vi IMproved

version 7.4.52
by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.alioth.debian.org
Vim is open source and freely distributable

Become a registered Vim user!

type :help register<Enter> for information

type :q<Enter> to exit

type :help<Enter> or <F1> for on-line help

type :help version7<Enter> for version info

0,0-1

All



Sublime Text

```
35 void base64_encode(const uint8_t * data, size_t len, char * dst,  
36 base64_charset variant)  
37 {  
38     const char * charset = (variant == base64_charset::URL_SAFE)  
39         ? URL_SAFE_CHARSET  
40         : STANDARD_CHARSET;  
41  
42     size_t src_idx = 0;  
43     size_t dst_idx = 0;  
44     for (; (src_idx + 2) < len; src_idx += 3, dst_idx += 4)  
45     {  
46         uint8_t s0 = data[src_idx];  
47         uint8_t s1 = data[src_idx + 1];  
48         uint8_t s2 = data[src_idx + 2];  
49  
50         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];  
51         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];  
52         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];  
53         dst[dst_idx + 3] = charset[(s2 & 0xff)];  
54     }  
55  
56     if (src_idx < len)  
57     {  
58         uint8_t s0 = data[src_idx];  
59         uint8_t s1 = (src_idx + 1 < len) ? data[src_idx + 1] : 0;  
60  
61         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];  
62         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];  
63         if (src_idx + 1 < len)  
64             dst[dst_idx++] = charset[(s1 & 0x0f) << 2];  
65     }  
66     dst[dst_idx] = '000';  
67 }  
68 }
```

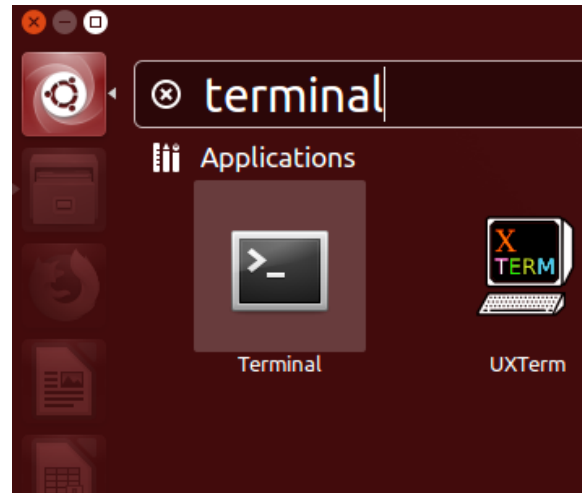
Build your program (cont.)

■ Command-line options in gcc

<code>--help</code>	print a description of the command-line options
<code>-o <file></code>	assign executable file's filename, default is "a.out"
<code>-c</code>	compile and assemble, but do not link
<code>-L</code>	add the search directory for libraries
<code>-I</code>	add the search directory for .h header files
<code>-l</code>	link with the specified library
<code>-O1</code> (-O2, -O3)	optimization levels
<code>-Wall</code>	display all warning messages

Build your program (cont.)

- Compile with gcc



```
test@test: ~/Documents
test@test:~$ cd Documents
test@test:~/Documents$ gcc -Wall helloworld.c -o helloworld
test@test:~/Documents$ ./helloworld
hello world !
test@test:~/Documents$
```



Outline

- **Environment**

- ☐ Install Ubuntu on VirtualBox
- ☐ Build your program

- **Background**

- ☐ What is a PID ?
- ☐ What a PID manager does ?
- ☐ Bitmap

- **Homework Assignment #1**

- **Reference**

What is a PID ?

- PID: process identifier
- Most operating systems identify processes according to a **unique** PID
 - UNIX
 - Linux
 - Windows
- Typically is an unsigned integer.
- Used as an index to access various attributes of a process within the kernel.

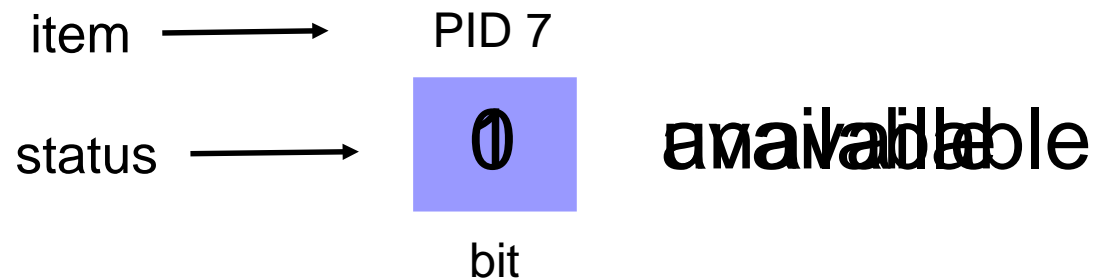
What a PID manager does ?

- An operating system's PID manager is responsible for managing process identifiers (PID).
- When a process is created :
 - It's assigned a unique PID by PID manager
- When the process completes :
 - The PID is returned to the PID manager
 - PID manager may re-assign the returned PID later
- No two active processes can have the same PID.

Bitmap

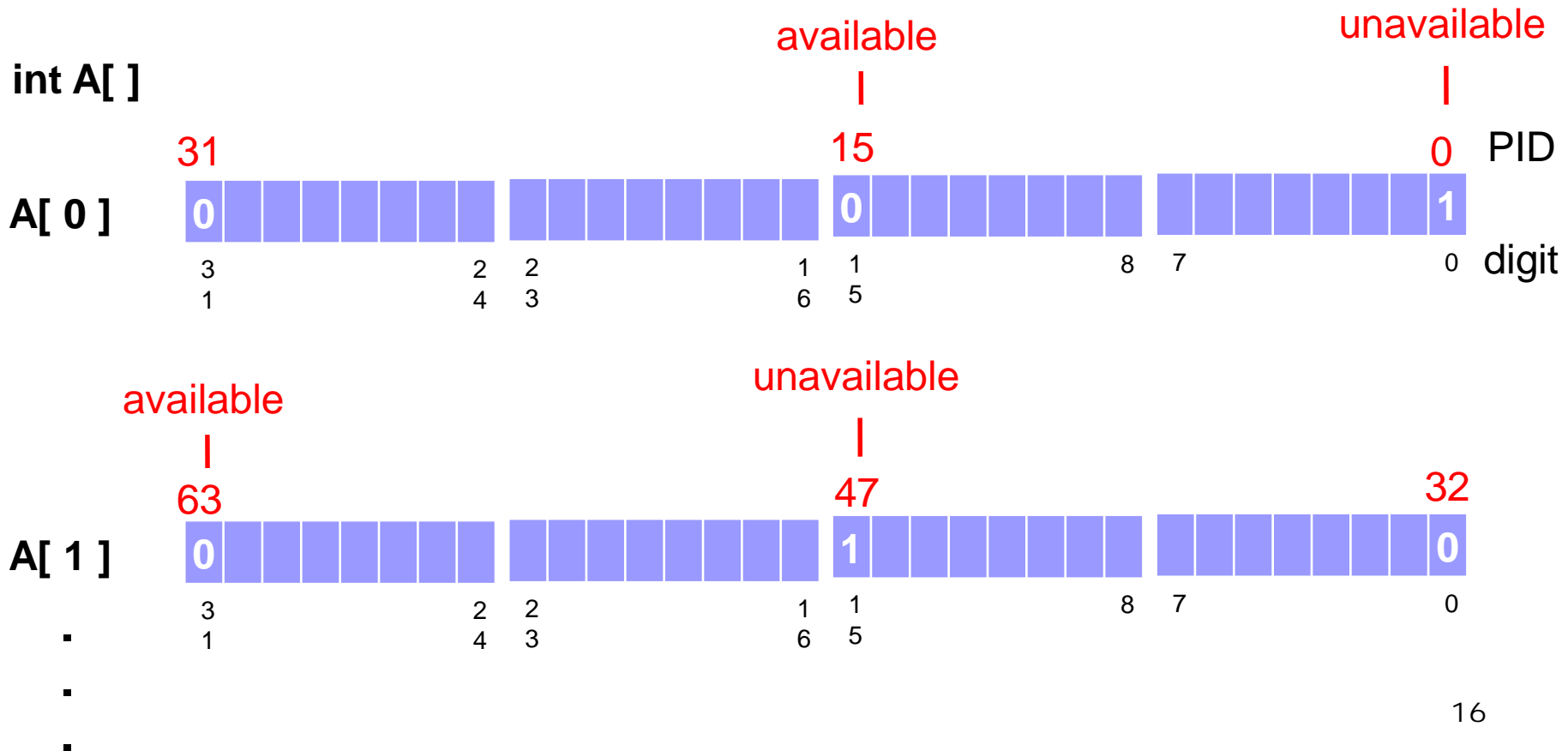
- A string of n binary digits are used to represent the status of n items.

Example :



Bitmap

- A string of n binary digits are used to represent the status of n items.



Bitmap (cont.)

- Bitmap is adopted in Linux.
- also known as **bit map**, **bit set**, **bit string**, or **bit vector**

- Advantage :

If we maintain 32,768 PIDs with

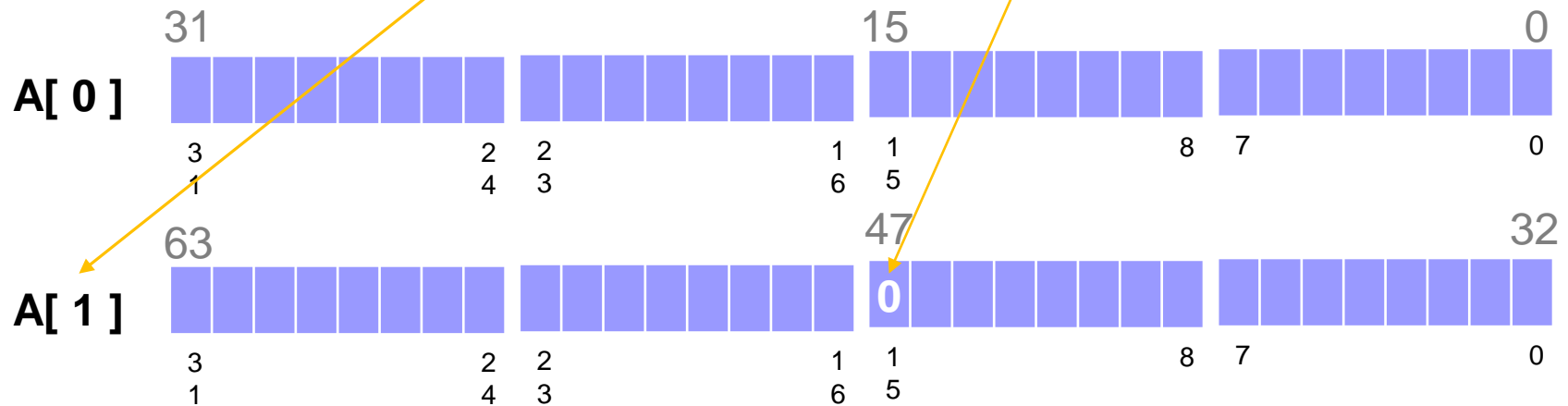
- Boolean array : 1 byte x 32768 = 32768 bytes, assume a PID uses 1 byte
- Bit array : 4 bytes x $\left\lceil \frac{32768}{32} \right\rceil = 4096$ bytes

Bitmap (cont.)

■ Bitmap operation

□ setBit(A, k)

$$A[(k/32)] \mid= (1 \ll (k \% 32))$$

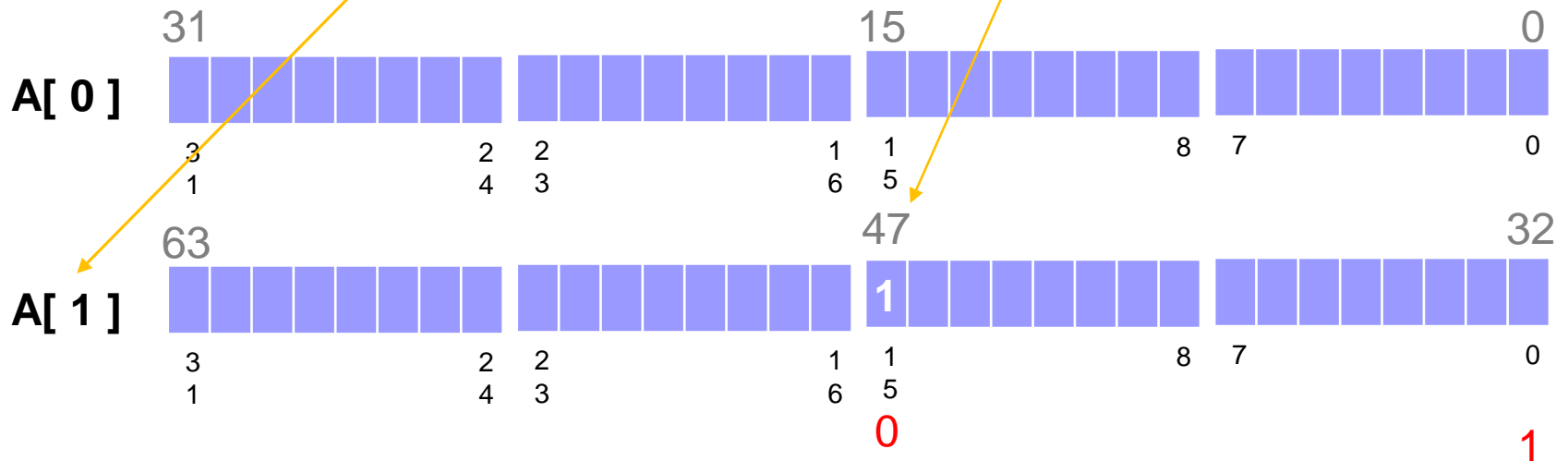


1

Bitmap (cont.)

□ clearBit(A, k)

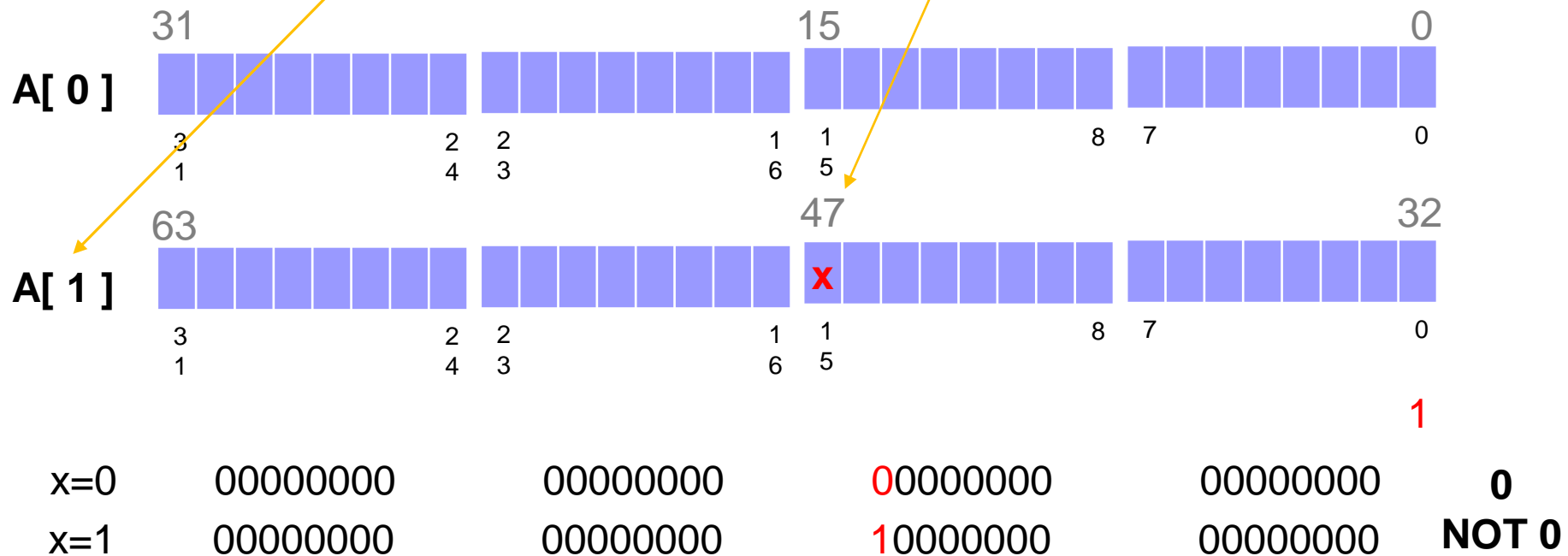
$$A[(k/32)] \&= \sim(1 \ll (k\%32))$$



Bitmap (cont.)

□ testBit(A, k)

$A[(k/32)] \& (1 \ll (k\%32))$





Outline

- **Environment**

- ☐ Install Ubuntu on VirtualBox
- ☐ Build your program

- **Background**

- ☐ What is a PID ?
- ☐ What a PID manager does ?
- ☐ Bitmap

- **Homework Assignment #1**

- **Reference**

Homework Assignment #1

- Implement PID manager which provides three APIs :
 - *int allocate map(void) :*
Creates and initializes a data structure for representing PIDs;
returns -1 if unsuccessful, 1 if successful
 - *int allocate pid(void) :*
Allocates and returns a PID; returns -1 if unable to allocate a PID (all PIDs are in use)
 - *void release pid(int pid) :*
Releases a PID

Homework Assignment #1 (cont.)

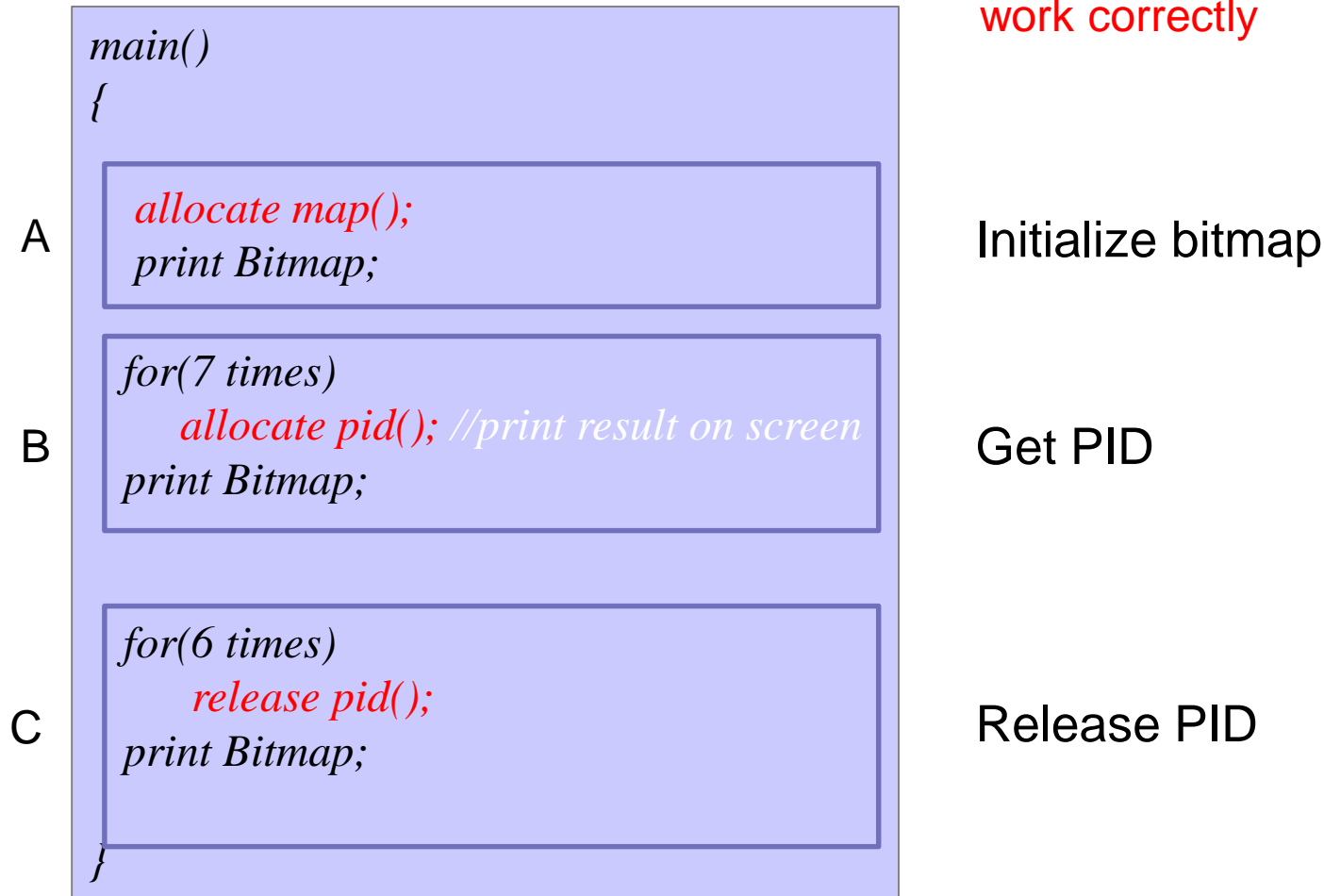
- Demands :
 - You should use a bitmap as a data structure to represent the availability of PIDs
 - Define PID range

```
#define MINPID 0
#define MAXPID 5
```

Homework Assignment #1 (cont.)

- Run your program as follow :

To prove your APIs
work correctly



Homework Assignment #1 (cont.)

■ Result :

```
### allocating bitmap ... ###
```

```
[ Bitmap ]
```

```
bitmap[0] = 0  
bitmap[1] = 0  
bitmap[2] = 0  
bitmap[3] = 0  
bitmap[4] = 0  
bitmap[5] = 0
```

Initialize bitmap and
show it

```
### allocating pid ... ###
```

```
process0 pid = 0  
process1 pid = 1  
process2 pid = 2  
process3 pid = 3  
process4 pid = 4  
process5 pid = 5
```

```
process6 pid = -1
```

```
[ Bitmap ]
```

```
bitmap[0] = 1  
bitmap[1] = 1  
bitmap[2] = 1  
bitmap[3] = 1  
bitmap[4] = 1  
bitmap[5] = 1
```

all of PIDs are unavailable,
so process6 get -1

```
### releasing pid ... ###
```

```
release process0 pid  
release process1 pid  
release process2 pid  
release process3 pid  
release process4 pid  
release process5 pid
```

```
[ Bitmap ]
```

```
bitmap[0] = 0  
bitmap[1] = 0  
bitmap[2] = 0  
bitmap[3] = 0  
bitmap[4] = 0  
bitmap[5] = 0
```

After releasing all PIDs,
Bitmap should be all zero

Turn In

- Deadline : 2019/ 10/ 23 P.M.11:59:59
- Upload to iLearning
- File name :
 - ☐ HW1_'ID'.zip (e.g. HW1_7108056057.zip)
 - ☐ Source code (hw1.c)
 - ☐ Word file
- If you don't hand in your homework on time, your score will be deducted **10** points every day.

Rules

- **No cheat work** is acceptable
 - You get zero if you copy other people's version
- **Only single job** is accepted

TA

- Name : Wei-Ming Su
- Email : g108056057@mail.nchu.edu.tw
- Lab : OSLab (1001)

Reference

- Textbook : Operating System Concepts, 10th Edition (chapter 3.3.1)

- PID :

<https://zh.wikipedia.org/wiki/%E8%BF%9B%E7%A8%8BID>

- Bitmap :

https://en.wikipedia.org/wiki/Bit_array

<http://www.mathcs.emory.edu/~cheung/Courses/255/Syllabus/1-C-intro/bit-array.html>

Reference

- Install & setup virtualbox :

<https://blog.xuite.net/yh96301/blog/66578586>

- Install Ubuntu :

<https://blog.xuite.net/yh96301/blog/432341564-VirtualBox+5.2%E5%AE%89%E8%A3%9DUbuntu+16.04>