Systems Programming
Introduction

What is systems programming?

- Programming where software is written to operate the hardware or interface with operating system
- Software produced by systems programming provides services to other software rather than providing services to end user in case of application software
- Where can we use systems programming? Its all around us
  - Operating Systems, Device Drivers
  - Game Engines – Xbox, PlayStation etc
  - Industrial Automation – Robotics, Control Systems, Monitoring Systems etc
  - Cloud – Infrastructure As A Service (IAAS), Platform As A Service (PAAS), Software As A Service (SAAS)
  - Internet of Things (IOT): Traffic Control Automation, Smart House, Air/Water Quality Control, Farming, Smart Grid/Meters and many more…

Difference between Application and Systems Software

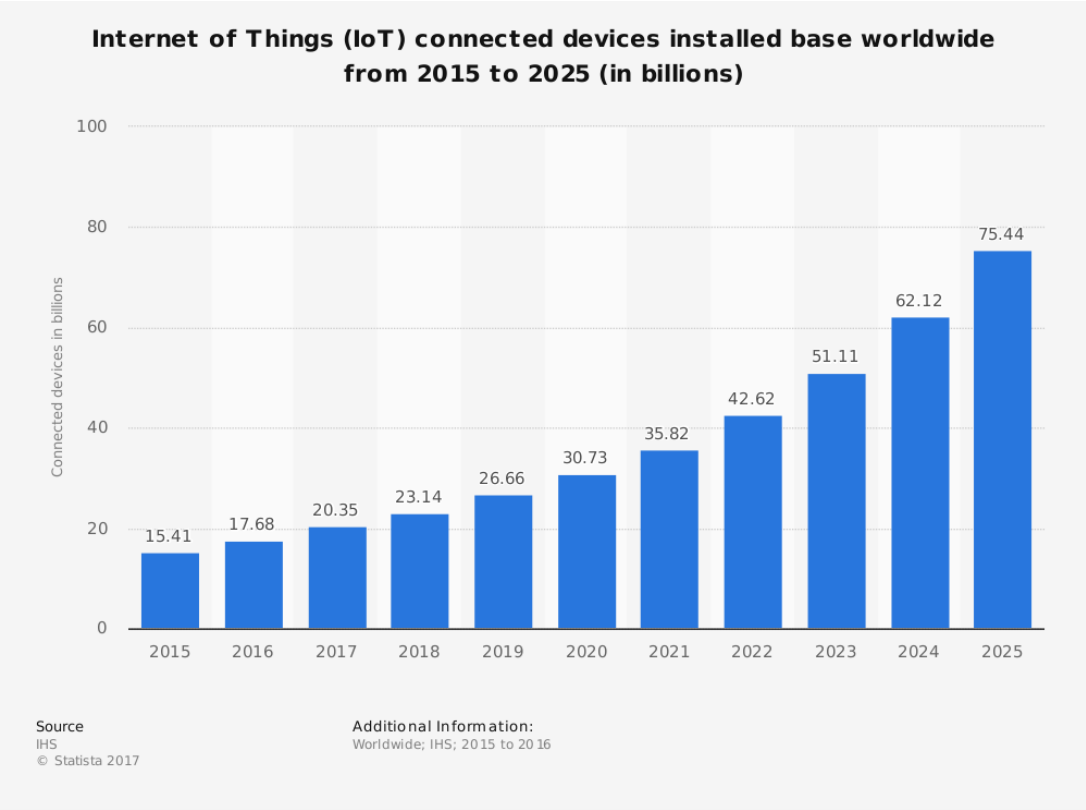
System Software

- System focused
- Requires detail knowledge of hardware and OS
- Provides services to other software such as application s/w
- Used to program hardware or interfaces with operating system
- Low level languages such as C/C++ and Assembly are used

Application Software

- User focused
- Does not require hardware knowledge
- Provides services to end user
- Used to provide functionality for a specific user application
- High level languages C#, Java, Python, HTML and so on are used

Why Systems Programming is important for Internet of Things (IOT)?



- Every device will need OS or Embedded Software to operate the hardware
- Almost every devices will require communication interface to share information with other devices
- Many devices will require sensing unit to be programmed
- Many devices will require logic to make decision based on the information received
- Many devices will require controller programmed to control the specific hardware unit

Source: https://easternpeak.com/blog/iot-ideas-for-business/

How IOT will be used for Smart City?

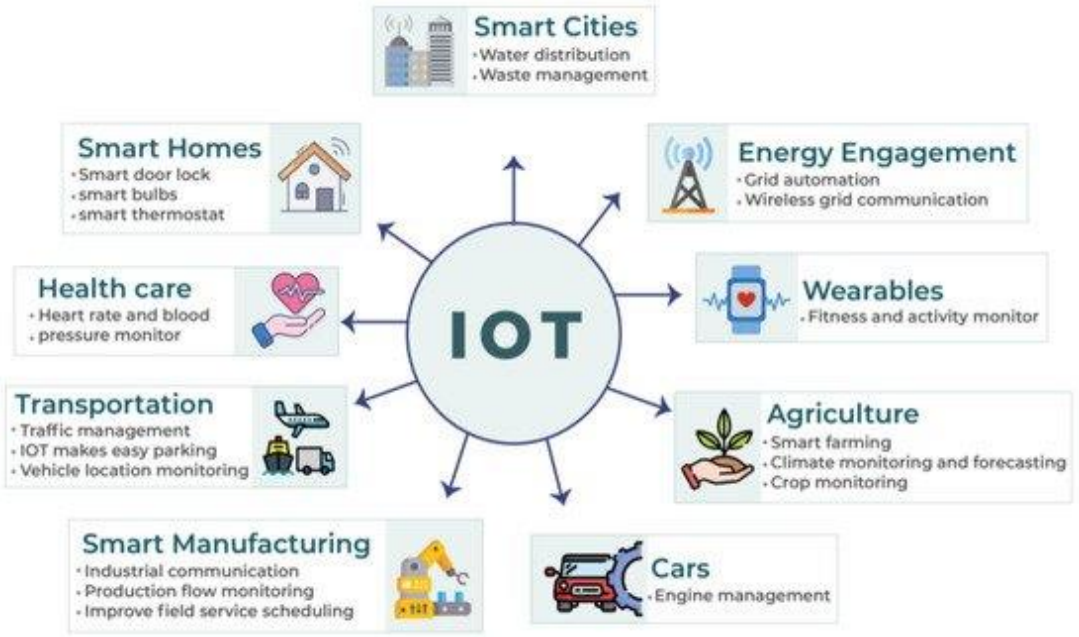
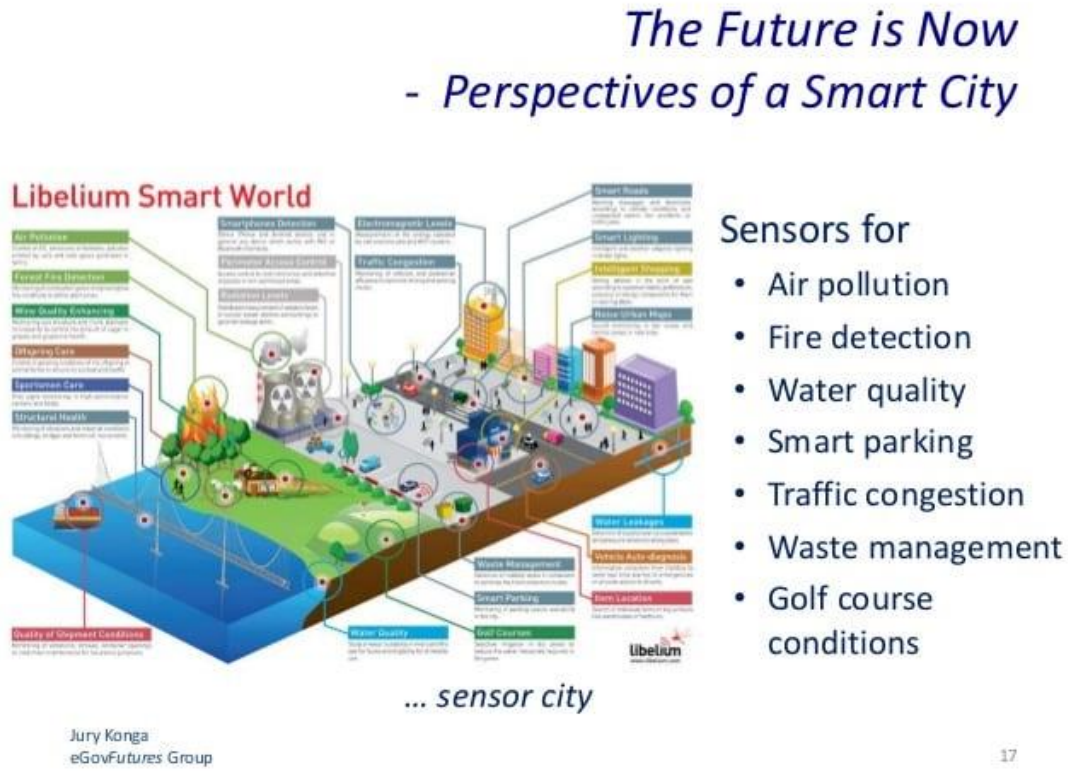
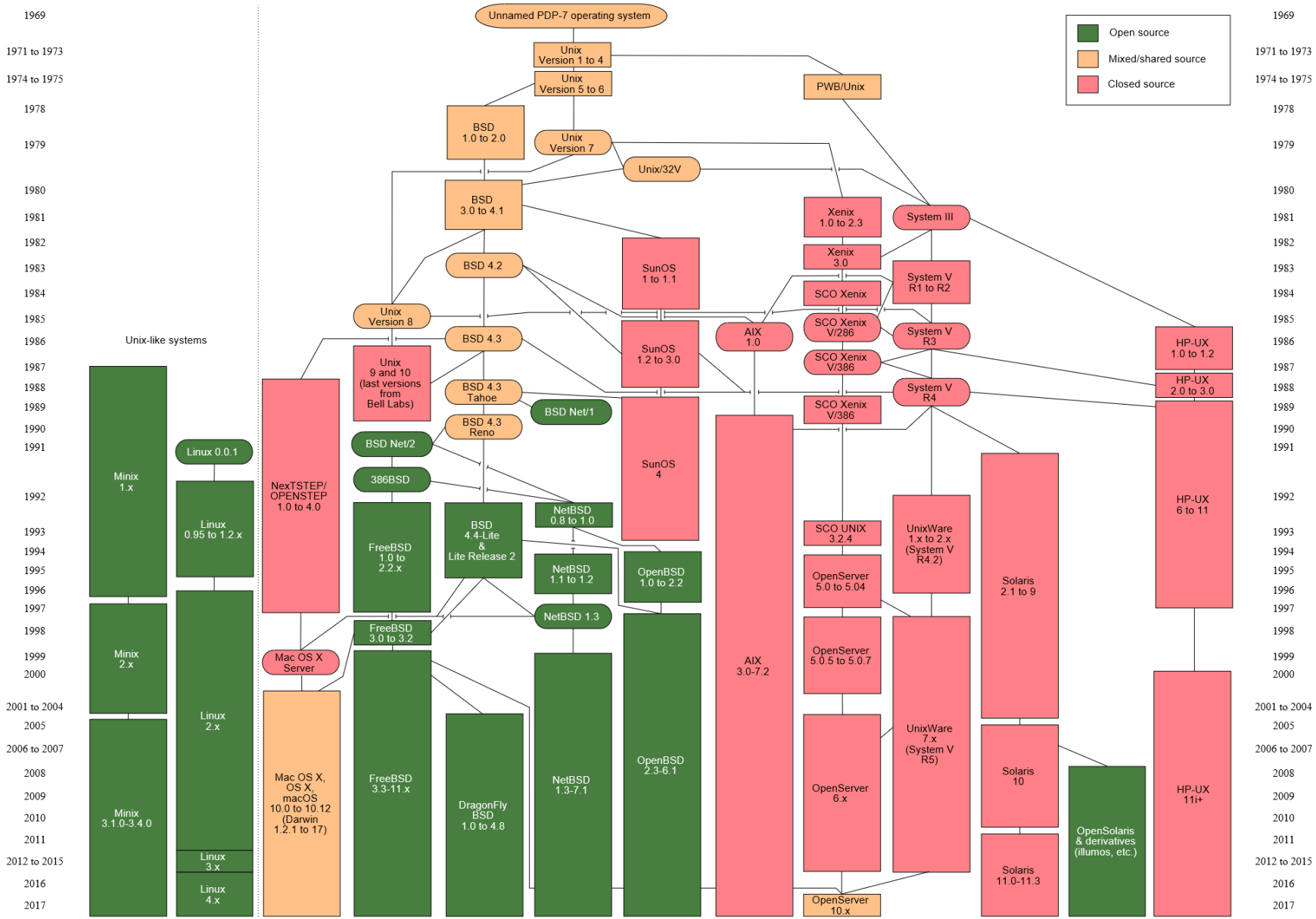


Image Source: https://www.mdpi.com/1424-8220/22/2/634/htm



Source: https://www.slideshare.net/JuryKonga/geosmart-cities-2020-beyond

We will be using Unix/Linux OS, why ?



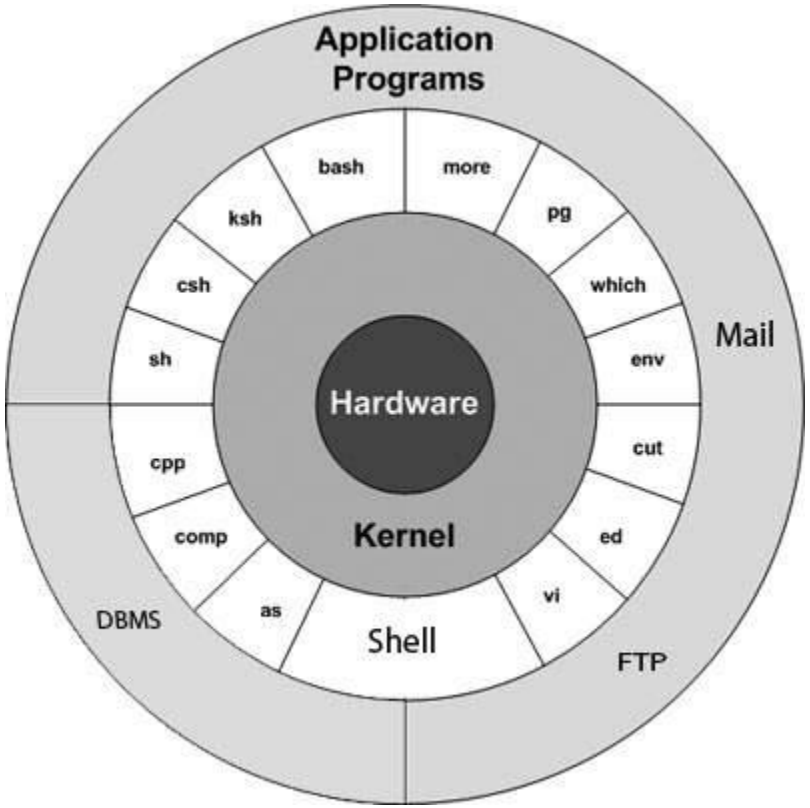
Source: https://en.wikipedia.org/wiki/UNIX_System_V

Unix flavors used today

- Apple iOS based on Darvin BSD, macOS is Unix-like
- Android based on Linux which is Unix like
  - Supported on many platforms :
    - DEC Alpha servers
    - ARM based embedded systems, mobile phones, tablets etc
    - Intel Skylake, Westmere, Nehalem, Xeon for desktops, laptops and servers
    - Apple-IBM-Motorola alliance PowerPC
    - Sun Microsystem (now acquired by Oracle corp) SPARC
- Playstation OS based on FreeBSD
- Ubuntu based on Linux
- Cloud OS is Unix-like
- And many more….

Unix/Linux Architecture

- Kernel – Heart of OS, interacts with hardware and performs tasks like memory management, process/thread/task scheduling, file management etc.
- Shell – Command line interface to run shell commands using variants C Shell, Bourn Shell, Korn Shell
- Commands and Utilities – 250+ standard commands and 3rd party utilities



Unix/Linux Basics

- How to logon?
  - Local computer → User ID and Password (Password is Case Sensitive)
  - Remote computer → Using ssh (Secure Shell) → userid@hostname or userid@IPAddress, enter password when prompted. You can also use ssh clients such as putty or MobaXTerm
- Current logged in user : whoami command
- Change Password: passwd command
- Getting help for commands and system functions: man command
- Text editors: gedit (GUI based), vi, vim, nano (non GUI)
- Compilers: gcc for c/c++, gfortran for fortran etc (we will be using gcc so get familiarity with it)

Vi editor regularly used commands

- vi filename
- a: enter insert mode, after the cursor
- i: enter insert mode, before the cursor
- O: enter insert mode, above the cursor
- o: enter insert mode, below the cursor
- r: replace one character under the cursor
- u: undo the last change to the file.
- x: delete character under the cursor
- yy: copy line
- dd: delete line
- :w: write
- :q: quit
- :q!: quit without saving changes
- /keyword : search for the keyword in text
- :n : go to line number n
- Vi tutorial: http://www.gnulamp.com/vi.html

Files and Directories

Directory Operations

- ls – list content of directory
- cd – change directory
- pwd – present working directory
- mkdir – create new directory
- rmdir – Remove/delete existing directory (and subdirectories)
- Single dot (.) and double dots (..) represents current directory and immediate parent directory respectively

File Operations

- cp – copy a file and/or directory
- scp – secure copy files and/or directory to another machine using ssh (i.e. file transfer)
- mv – rename a file and/or directory or move file from one directory to another
- rm – remove/delete file
- cat, head, tail, more, less, wc : explore yourself

Files and Directories

- File Permissions: rwx (read, write and execute) for user, group and others e.g.
  - -rwxrw-r-- 1 linaro linaro 4044 Aug 29 08:13 /home/linaro/.bashrc
  - Indicates linaro user can read,write and execute, users from group linaro can read and write but can’t execute; and other users who are not in linaro group can only read
- Change file permission: chmod command (using bit pattern e.g. 766 indicates rwx for user, rw- for group and rw- for other user)
- Change file ownership : chown command
- Executing commands as superuser (mostly root): sudo command

Standard Unix file streams

- Standard Input STDIN with file descriptor 0
- Standard Output STDOUT with file descriptor 1
- Standard Error STDERR with file descriptor 2

- Redirection of standard output > or >> (> : create file, >> : append to existing file)
  - ls –ltr > dirlist.txt
  - ls –ltr >> dirlist.txt
- Redirection of standard input <
  - E.g. cat < dirlist.txt

Processes

- ps – lists currently active user processes
- ps -aux or ps -ef – lists all active processes in long format
- kill n – kill process with process id = n
- kill -9 n – force to kill process id = n

- Keys CTRL-z – force process to move to background
- fg – bring process to foreground

- top – provides system utilization information
- time – calculate time for a given command

Other Unix/Linux commonly used commands

- awk – pattern scanning and processing language

Print the total number of kilobytes used by files in the current directory

ls -l . | awk '{ x += $5 } ; END { print "total kilobytes: " (x /1024) }‘

Print sorted list of login names

awk -F: '{ print $1 }' /etc/passwd | sort

Print only elements from column 2 that match pattern using stdin

awk ' /'pattern'/ {print $2} '

- grep – search for files with a given pattern

Search /etc/passwd file for the user harry

grep harry /etc/passwd

Search all files in current directory and subdirectory for expression “harry potter” with ignore case

grep -r -i “harry potter” /home/joe/

Other Unix/Linux commonly used commands

- find – search for files in directory hierarchy

- sed – stream editor for filtering and transforming text

Replace every occurrence of Nick with John in report.txt

```
sed 's/Nick/John/g' report.txt
```

Add 8 spaces to the left of a text for pretty printing.

```
sed 's/^/ /' file.txt >file_new.txt
```

Write all commands in script.sed and execute them

```
sed -f script.sed file.txt
```

Print only lines with three consecutive digits

```
sed '/[0-9]\{3\}/p' file.txt
```

Other Unix/Linux commonly used commands

- cut – remove section from each line of file
- ping – sends echo request to host/ip address
- telnet/putty – tool used to connect to remote host using ssh from non-unix based machine
- uname – print system information
- lsb_release – print distribution specific information
- locate – Locate files by name

What is profile?

- For each type of shell when you login default profile file gets executed
- e.g. for Bourne Again Shell (bash)
  - .bashrc file in HOME folder (first dot in the file name indicates hidden file) gets executed