# Topic 1: Basic Concept

## Overview

### Guidance

Operating System Concepts 10th Edition.pdf chapter 1, chapter 20.1-20.3

Advanced Programming in the UNIX Environment, 3rd Edition.pdf 1.6 2.2 2.3

wiki:[History of Unix](https://en.wikipedia.org/wiki/History_of_Unix), [Linux distribution](https://en.wikipedia.org/wiki/Linux_distribution)

### Practice

1. Describe the following concepts

DMA, POSIX, Timer

1. How does the application program using the OS resources? What OS resources are there?
2. What types of operating systems exist? What are their characteristics?
3. What is the relationship between unix and linux? What is the relationship between redhat and linux? What is the defference between linux destribution and linux kernel?
4. What is the relationship between program and process?
5. Operating System Concepts 10th Edition.pdf chapter1 -> Practice Exercises 1.1
6. Operating System Concepts 10th Edition.pdf chapter1 -> Practice Exercises 1.3

## System Call

### Guidance

Operating System Concepts 10th Edition.pdf Chapter 2 System Calls

Advanced Programming in the UNIX Environment, 3rd Edition.pdf 1.11

ioctl dup

ulimit -a, What does these content mean? How to change it

### Practice

Write a program for each type of system call using the function in page 68.

If it can be done early by you, to explore more system calls and put the cases in your programs

## Process

### Guidance

Operating System Concepts 10th Edition.pdf 3.1-3.3

Operating Systems - Internals and Design Principles 7th.pdf 3.1-3.3

* What are similarity and difference between parent process and child process?
* ps/pstree
* Process state switch

### Practice

1. Run a program try to find it in pstree
2. What is the meaning of each filed of command ps -elf?
3. How many ways to create, suspense or terminate a process? Illustrate
4. Why do we need a PCB? What informations are stored in the PCB? What role does the PCB play when a process is created/scheduled/executed?
5. In what scenario switch between different states of a process？Illustrate

## Thread

### Guidance

Operating Systems - Internals and Design Principles 7th.pdf 4.1-4.2

Operating System Concepts 10th Edition.pdf 4.3-4.4

### Practice

1. For multiple threads in the same process, which resources are shared and which resources are unique?
2. What is the thread type created in Linux using pthread\_create? ULT or KLT?
3. What are the advantages and disadvantags for multi-process(only one thread in each process) and multi-thread do programming?
4. What is reentracy functions? What is non-reentrant functions? Illustration
5. Can multi thread share a routine function start\_routine? What should we pay attention to if we use it this way
6. How to make a thread never end?

# Topic 2: IPC

## Basic Concept

### Guidance

unix network programming Vol.2.pdf chapter 1-3

Advanced Programming in the UNIX Environment, 3rd Edition.pdf chapter 15

* The life cycle of the IPC object
* What is different between Posix and System V IPC?
* Can these ways be used for inter-thread communication

### Practice

1. According to your understanding, try to divide all IPCs into different categories, and the reason
2. To summarize, what are the characteristics of each IPC?
3. Which can be used for remote communication and which can be used for local communication. Can remote ways be used for local communication?

## Data Passing

### Guidance

unix network programming Vol.2.pdf part 2

Advanced Programming in the UNIX Environment, 3rd Edition.pdf 17

### Practice

1. Use fork/exec simulation executive "ping 127.0.0.1 -c 3" and get the result
2. Follow the client/server model, client sends a filename to server, response its content, implement by unix socket
3. Follow the producer/consumer model, producer sends message MSG1/MSG2/MSG3 to consumer, print the messages when the consumer received it. Give a sample for the 2 kinds of message queue separately

## Synchronization & Mutual Exclusion

### Guidance

unix network programming Vol.2.pdf 7 10 11

mutexes, condition variables, read-write locks, semaphores

### Practice

Use semaphores(system v or posix) to achieve synchronization and mutual exclusion respectively

1. Write a program synchronizes by semaphore
2. Write a program that is mutually exclusive by semaphore
3. Build several scenarios that generating deadlocks and implement them programmatically