

Linux System Programming

Using C

Sarath M

Listings

2.1	programs/getpid.c	10
2.2	programs/system.c	10
2.3	programs/system2.c	11

Contents

1	Introduction	7
1.1	Topics	7
1.2	Defenition's	7
1.3	OS in Embedded Systems	7
1.4	Components of an OS	8
2	Process Management	9
2.1	Process Manager	9
2.2	PID and PPID	10
2.3	System() function	10
2.4	Remarks	11
2.5	Corollaries	11
2.6	Propositions	11
2.6.1	Several equations	11
2.6.2	Single Line	11
2.7	Examples	11
2.7.1	Equation and Text	12
2.7.2	Paragraph of Text	12
2.8	Exercises	12
2.9	Problems	12
2.10	Vocabulary	12
3	Presenting Information	13
3.1	Table	13

3.2	Figure	13
	Bibliography	15
	Books	15
	Articles	15
	Index	17

1. Introduction

1.1 Topics

1. Process Management
2. File and file management
3. Memory and memory management
4. Signals and Signal handling
5. Thread and Thread management
6. Inter Process Communication
7. Process Synchronisation
8. Shell Scripting

1.2 Definition's

Operating System OS is a resource manager/allocator (rather than a mere interface between the user and hardware) which is responsible for managing the resources which is connected to the CPU

BIOS Basic Input Output Systems When we switch ON the system, BIOS program is executed. First job of BIOS is to check if basic input output are connected or not. After that the BIOS executes a program called Bootloader

Bootloader picks up an OS from hard disk and loads it into RAM. Load time/Boot time is the time taken for this operation.

Grub loader Boot loader in Linux

NT loader Boot loader in Windows

1.3 OS in Embedded Systems

Why OS in embedded systems?

Without OS only one program can be run at a time

With OS multiple process can be run simultaneously. Thus performance of the product increases.

1.4 Components of an OS

1. Application
2. Services

Application are optional services. Applications runs only when we intentionally run it.

Services are mandatory. Kernel starts executing when OS is loaded into RAM

Kernel All the services combined is called a kernel

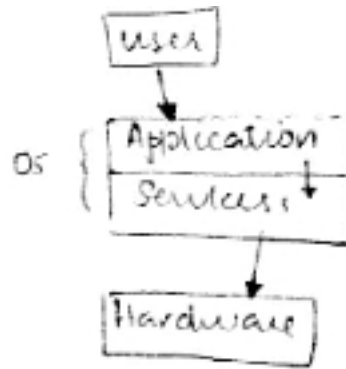


Figure 1.1: OS

2. Process Management

Process Thre program which is in execution is called as the process. To execute a file say a.out, a copy of a.out is loaded into RAM

Process Manager - Manages the different process.

Commands

ps -e Command to display the processes which are currently running

2.1 Process Manager

For every process, the process manager will provide a process ID, ie; the process manager identifies each process with a process ID.

Commands

- *./a.out &* : If we want to run a command in background
- *fg* : command to bring the background process to foreground
- *fg <jobId>*: To move a specific process to foreground
- *ps -e | grep pts/0*: If we want to list all programs running in terminal
pts/0

1821 [1]
↓
process id
ls given by manager
Job id - given by terminal.

Figure 2.1: Output

Shell - Command Interpreter. When user wants to interact with the OS we use the shell. **ex: bash**

Commands

kill: - command used to send signals to a particular process

ex: `kill -9 1769`, where 1769 is process id

2.2 PID and PPID

Listing 2.1: programs/getpid.c

```

1 #include <sys/types.h>
  #include <unistd.h>
3 #include <stdio.h>

5 void main()
  {
7     printf("Hello!\n pid = %d, ppid = %d\n", getpid(), getppid()
        );

9     printf("Waiting (while(1))\n");
      while(1);
11 }

```

Commands

- **get_pid():** returns the process id
- **get_ppid():** return the parent process id

The parent is nothing but the bash. Whenever a new terminal is opened, a bash process is created. This bash is the parent of the program being executed

2.3 System() function

Listing 2.2: programs/system.c

```

1 #include <stdlib.h>
  #include <stdio.h>
3
5 int main()
  {
7     printf("Hello\n");
      system("ls");
      printf("Hi\n");
9 }

```

Listing 2.2: Ouput

```

Hello
a.out getpid.c system.c system.txt
Hi

```

2.4 Remarks

This is an example of a remark.

R The concepts presented here are now in conventional employment in mathematics. Vector spaces are taken over the field $\mathbb{K} = \mathbb{R}$, however, established properties are easily extended to $\mathbb{K} = \mathbb{C}$.

2.5 Corollaries

This is an example of a corollary.

Corollary 2.5.1 — Corollary name. The concepts presented here are now in conventional employment in mathematics. Vector spaces are taken over the field $\mathbb{K} = \mathbb{R}$, however, established properties are easily extended to $\mathbb{K} = \mathbb{C}$.

2.6 Propositions

This is an example of propositions.

2.6.1 Several equations

Proposition 2.6.1 — Proposition name. It has the properties:

$$||\mathbf{x}|| - ||\mathbf{y}|| \leq ||\mathbf{x} - \mathbf{y}|| \quad (2.1)$$

$$||\sum_{i=1}^n \mathbf{x}_i|| \leq \sum_{i=1}^n ||\mathbf{x}_i|| \quad \text{where } n \text{ is a finite integer} \quad (2.2)$$

2.6.2 Single Line

Proposition 2.6.2 Let $f, g \in L^2(G)$; if $\forall \varphi \in \mathcal{D}(G)$, $(f, \varphi)_0 = (g, \varphi)_0$ then $f = g$.

2.7 Examples

This is an example of examples.

2.7.1 Equation and Text

■ **Example 2.1** Let $G = \{x \in \mathbb{R}^2 : |x| < 3\}$ and denoted by: $x^0 = (1, 1)$; consider the function:

$$f(x) = \begin{cases} e^{|x|} & \text{si } |x - x^0| \leq 1/2 \\ 0 & \text{si } |x - x^0| > 1/2 \end{cases} \quad (2.3)$$

The function f has bounded support, we can take $A = \{x \in \mathbb{R}^2 : |x - x^0| \leq 1/2 + \varepsilon\}$ for all $\varepsilon \in]0; 5/2 - \sqrt{2}[$. ■

2.7.2 Paragraph of Text

■ **Example 2.2 — Example name.** Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. ■

2.8 Exercises

This is an example of an exercise.

Exercise 2.1 This is a good place to ask a question to test learning progress or further cement ideas into students' minds. ■

2.9 Problems

Problem 2.1 What is the average airspeed velocity of an unladen swallow?

2.10 Vocabulary

Define a word to improve a students' vocabulary.

Vocabulary 2.1 — Word. Definition of word.

3. Presenting Information

3.1 Table

Treatments	Response 1	Response 2
Treatment 1	0.0003262	0.562
Treatment 2	0.0015681	0.910
Treatment 3	0.0009271	0.296

Table 3.1: Table caption

3.2 Figure

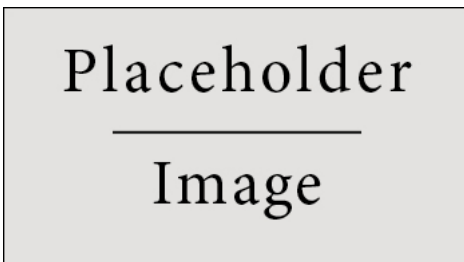


Figure 3.1: Figure caption

Bibliography

Books

Articles

Index

C

Corollaries 10

E

Examples 11
 Equation and Text 11
 Paragraph of Text 11
Exercises 11

F

Figure 15

G

getpid() 10
getppid() 10

P

Problems 11
Process Management 9
Propositions 11
 Several Equations 11
 Single Line 11
ps command 9

R

Remarks 10

S

system() 10

T

Table 15

V

Vocabulary 12