Request for public comments

OSS Model Curriculum for Software Engineering Education
- Skill-sets and Sample curriculum -

Draft 1.0

April 24, 2009

Northeast Asia OSS Promotion Forum WG2

China OSS Promotion Union
Japan OSS Promotion Forum
Korea OSS Promotion Forum

Table of Contents

Foreword

The NEAOSS Forum (Northeast Asia Open Source Software promotion Forum) was formed by the Chinese, Korean and Japanese governments and regional organizations for OSS promotion; the China OSS Promotion Union, Korea OSS Promotion Forum and Japan OSS Promotion Forum. The Forum intends to promote Open Source Software within the northeast Asian area.

The NEAOSS Forum formed “WG2: Human Resource Development” in order to promote and develop Human Resources for Open Source Software application and development in Jul. 2004.

In accordance with the CJK DG Memorandum for the 4th (Tianjin), 5th (Fukuoka) and 6th (Seoul) NEA OSS Promotion Forums, the WGs have been discussing topics such as mutual testing and certification of OSS experts, and the development of curriculum and textbooks for OSS developers and users.

The NEAOSS Forum WG2 formed a subsidiary group, Task Force 1, in Apr. 2006 at the NEA OSS Promotion Forum in Tianjin. It addresses research of the current status of Human Resource Development for Open Source Software in the northeast Asian region on both the demand and provider side and development of the NEA HRD Model Curriculum for colleges, lecturers, enterprises and institutes to develop human resources with OSS skills.

WG2 delivered the “NEA HRD Analysis Report" (draft 1.0) in December 2007 and published the report Ver.1.0 by September 2008. WG2 develops the NEA HRD Model Curriculum for OSS which is based on this research result.

As far as open source software (hereinafter called “OSS”) is concerned, there has been steady improvement to date in operating systems, middleware, networks, and development tools to build IT systems, thus reinforcing the status of OSS as a consistent software technology infrastructure to sustain our information economy. However, while there are growing needs for OSS as an important infrastructure, there is a severe shortage of engineers competent enough to use OSS, which is one of the greatest factors impeding the spread of the software. The Survey on NEA OSS human resource development titled the “NEA HRD Analysis Report” pointed out a wide gap among businesses and IT service providers between expectations for OSS engineers and the real situation.

For OSS to achieve wide recognition for its usefulness and to spread throughout the society, it is essential to cultivate IT engineers capable of building, operating, and maintaining OSS-based IT systems with sufficient knowledge of OSS. For that purpose, we are urged to establish a model curriculum to be used as a reference in OSS technical training and to promote it among universities, vocational schools, IT training providers, and other organizations. Under these circumstances, this project resulted in drafting of the NEA HRD Model Curriculum.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. NEAOSS shall not be held responsible for identifying any or all such patent rights.

Contributors

Korea OSS Promotion Forum

Kern Koh, Seoul National University
Doohyun Kim, Konkuk University
Minsuk Lee, Hansung University
Sang-hyo Song, BIZ Kernel, Inc.
Soon Young Jung, Korea University
Teak-Jong Kim, TmaxSoft
Je-Ho Kim, CDC Group, Inc.
Ki-Cheol Han, Darwin Consulting and Solution
Young Jong Kim, Opensource Community Lab.

Japan OSS Promotion Forum

Hiroshi Miura, NTT DATA Corporation
Ryosuke Kajiyama, Sun Microsystems, Inc.
Kazuo Hiyane, Mitsubishi Research Institute
Kennichiro Hamano, NTT DATA Corporation
Makoto Oya, Shonan Institute of Technology
Koichi Oouchi, Information-processing Promotion Agency

China OSS Promotion Union

Chen Zhong, Peking University
Huiping Sun, Peking University
Guo Rui Wu, Peking University

1. Introduction

1.1 Terminology and definitions

(1) Skills

In general, "skill" is often referred to as knowledge and expertise on element technology of a specific product, know-how on the application of service, and specific programming language etc. In the field of OSS, similarly skill means expertise and knowledge for following diverse, but not limited, activities:

- Creating and maintaining new OSS projects
- Updating existing OSS as contributors or committers
- Applying OSS to software development/system development
- Utilizing OSS for system construction/ management

(2) Skill Category

Skill category is a set of skills under the same technical domain. For example, Java, C/C++ and web programming skills are element skills under the programming skill category. Table 1 shows OSS skill sets composed of 37 skills under 9 skill categories dealt with in this document.

(3) Topic, subtopic, and unit

A skill is defined by topics. A topic has subtopics. The topic is like a chapter in a textbook. For example, “building a cross development environment” is a topic under the “embedded S/W development environment” skill. Meanwhile, “able to use minicom” is a subtopic under the “building a cross development environment” topic.

The subtopics are grouped into three levels, but a topic does not need to have subtopics in all three levels. A group of subtopics in a level is called a unit. The unit is described with objective, prerequisite and unit code as shown in Figure 1; consequently, it contains the most detailed items supporting a skill.

In some cases, a certain unit is equivalent to or inclusive of the other unit in a different topic in a different skill. Such unit can be shared with the other corresponding unit. For example, the level-1 unit in the “kernel debugging” topic, whose subject is “Setup remote debugging environment and debug“, in the “device driver development” skill can be included in the level-1 unit in the “remote debugging” topic of the “Embedded S/W Development Environment” skill.

Table 1. OSS skill set (9 Skill Categories and 37 Skills)

Skill Categories	Skills	Skill Categories	Skills
Basic	Knowledge of OSS	Multimedia System	Multimedia Programming
	Legal Affairs		Multimedia Service Platform
	Computer System and Architecture	Development system	Development Frameworks
	Distributed Architecture		Development Tools
System	Concept of Linux and Basic Operations		Integrated Development Env.
	Kernel of Linux	Security	Encryption
	Linux System Management		Network Security
	Linux System Programming		OS Security
	Network Server Management	RDB	Basic Skills in RDB
	Cluster System Architecture		RDB system management

	Concurrent System Programming		
	Java EE Application Server (former WAS)	Embedded SW	Embedded System
Network	Linux Network Programming		Embedded Development Env.
	Network Architecture		Embedded Application Development
	Network Management		Embedded System Optimization
Programming	Java		Device Driver Development
	C		
	C++		
	Light Weight Language		
	GUI		
	Web Programming		

SKILL CATEGORY NAME	〈 Name of Skill Category 〉		SKILL CATEGORY NO.	〈N〉
SKILL NAME	〈 Name of the Skill 〉		SKILL NO.	〈M〉
TOPICS	LEVEL	DESPRIPTIONS & SUBTOPICS		Topic Code
Topic 1 (has a level II topic with prerequisite 〉	I	-		-
	II	Objective	〈 Descriptive objective of the level of topic 1 〉	N-M-1-II
		Prerequisite	〈 prerequisite topic code(s) 〉	
		- Subtopic 1 ■ Sub-subtopic . ■ … - Subtopic 2 ■ …		
	III	-		-
Topic a 〈 has a code shared level I topic, and a level II topic with no prerequisite 〉	I	Objective	〈 Copied objective from the shared topic P-Q-r-I 〉	CODE SHARE P-Q-r-I
	II	Objective	〈 Descriptive objective of the level of topic a 〉	N-M-a-II
		- Subtopic 1 ■ Sub-subtopic . ■ … - Subtopic 2 ■ …		
	III	-		-
Topic b 〈 Code shared topic Level I with X-Y-w-I, Level II with X-Y-w-II〉	I	Objective	〈 copied objective from the shared topic X-Y-w-I 〉	CODE SHARE X-Y-w-I
		Prerequisite	〈 copied prerequisite codes from shared topic X-Y-w-I 〉	
	II	Objective	〈 copied objective from the shared topic X-Y-w-II〉	CODE SHARE X-Y-w -II
	III			-

Figure 1. Skill definition form

(4) Skill proficiency and level

Skill proficiency is defined as a necessary degree of maturity for performing corresponding activities. Skill proficiency is described in the criteria as “able to do so,” and it is an evidence of reaching a certain level of capability in a specific activity. In other words, it provides a guideline whereby an individual who is assessed at a specific level in an activity is able to perform a certain degree of work for a corresponding particular skill. The OSS human resource is indexed by three levels as follows:

- Level-1: As a professional, he/she understands minimum common knowledge for information technology. He/she can work by utilizing IT knowledge for his/her job.
- Level-2: Working to realize an IT solution, product and service. He/she plays a role with advice from a senior person.
- Level-3: Working to realize an IT solution, product and service, he/she fulfills a role on his own.

1.2 Coding rules

For the convenience in utilizing skill definitions, the unit shall have a unique code following simple and consistent rules. In this document, the unit is coded with the form of C-S-T-U, each element of which stands for the following:

- C: skill Category
- S: Skill
- T: Topic
- U: Unit

In the previous example, the “able to use minicom” is one of the subtopics in the Level-1 unit coded as 9-2-4-I, because the “Embedded SW” skill category, the “embedded S/W development environment” skill, and the “building a cross development environment” topic are numbered as 9, 2, and 4, correspondingly. Meanwhile, in the case of sharing a unit with another topic, the code of the corresponding unit is used. For example, the Level-1 unit of the “kernel debugging” topic can have the same code as this 9-2-4-I, because the objective is the same.

1.3 Overall model curriculum architecture

As shown in Figure 2, the curriculum is a set of subjects which have precedence relationships, and a subject is a collection of units in a topic. For example, while subject 2 of curriculum 2 is organized with only the Level-2 unit (1-2-1-II) of topic number 1-2-1, subject 1 of curriculum M is composed of two units, i.e., 1-2-1-II and 1-2-1-III. Whether a unit is contained in the subject of a certain curriculum or not depends on the curriculum designer’s purpose. Similarly, by including objectives, outcomes, prerequisites, modules and hours in the syllabus, the organization of a subject is wholly dependent on the curriculum designer.

On the other hand, subjects can be represented by corresponding topics because the skill sets can be applied to a curriculum with scalability. While Figure 2 shows a case of a curriculum with subtopic-level granularity, Figure 3 shows a case with topic-level granularity.

The curriculum designer is typically a group of persons, as well as one person, from industrial sectors, public educational and training sectors, and/or in-house training sectors. They may firstly survey the status of the OSS human resource demands in a specific job category, and seek a solution by designing curriculum properly to meet the survey demands.

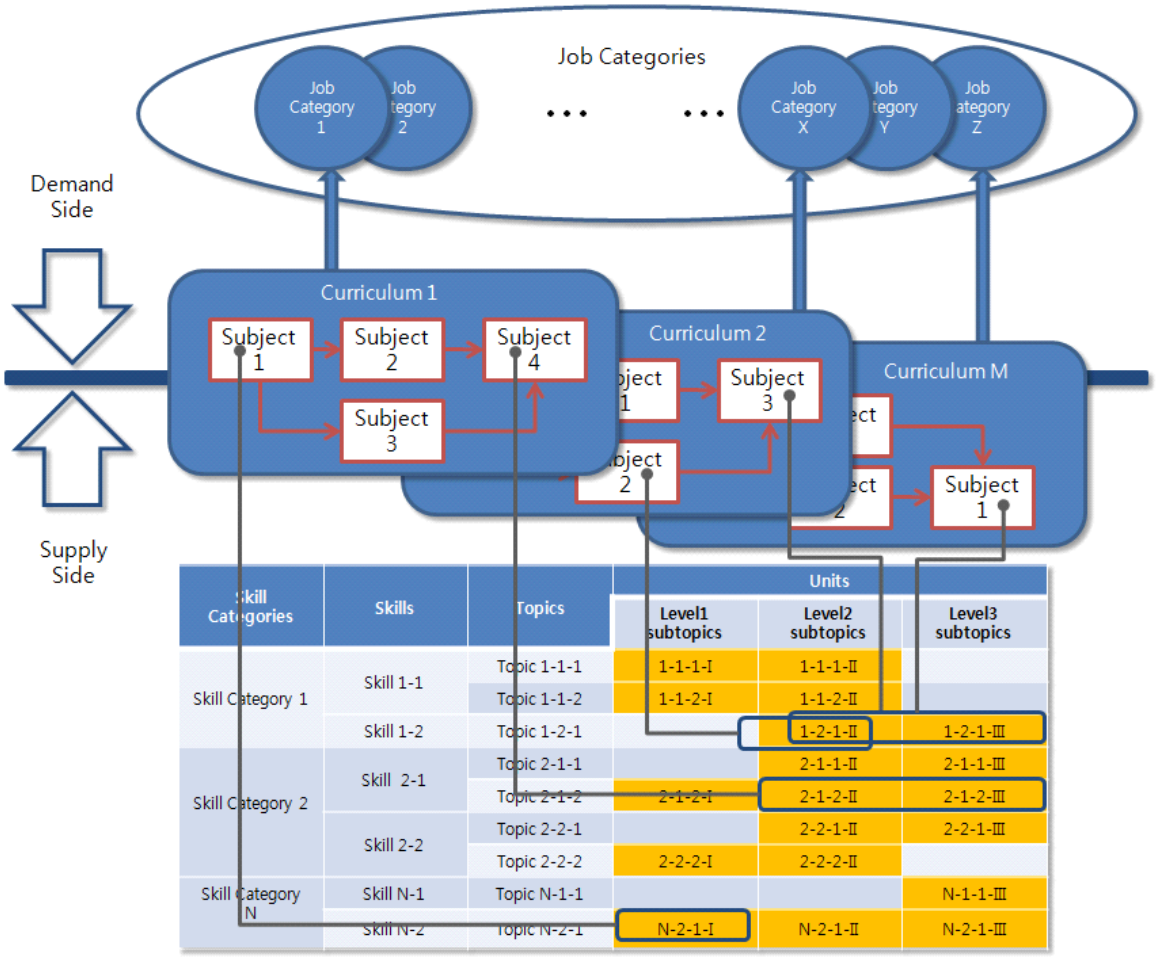

Figure 2. Subtopic-based relationships among skills, subject, curriculum and job categories

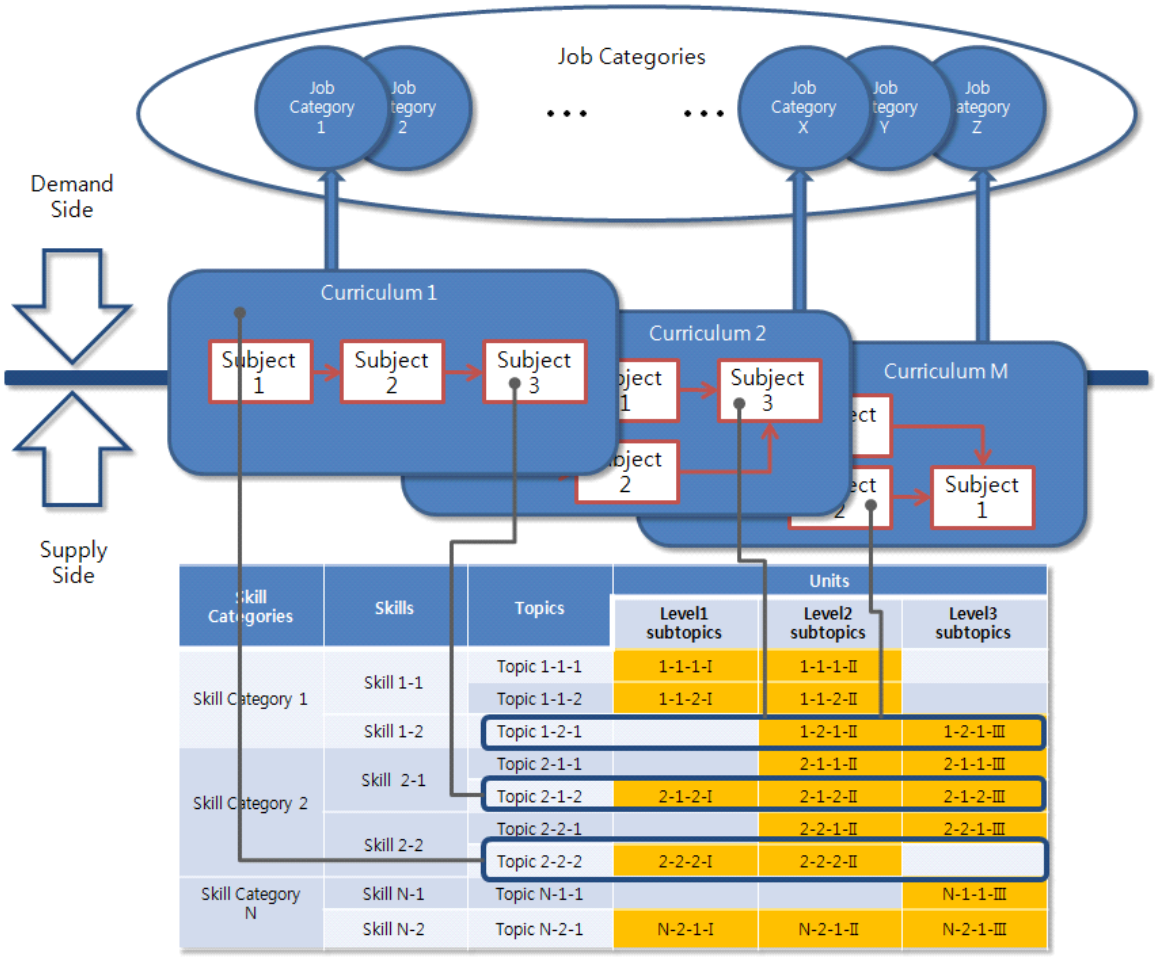

Figure 3. Topic-based relationships among skills, subject, curriculum and job categories

1.4 Industrial Masking Sheet & Personal Record Sheet

In the course of employment interviews, a company may need indicators to assess the maturity of a prospective employee. In this document, the industrial masking sheet is introduced as an OSS skill maturity assessment tool. As shown in Figure 4, an industrial masking sheet, that is, “masking sheet” in short, is prepared by a group of companies or a single company in a job category in order to represent the skill specifications for the human resources whom they are seeking. The masking sheet can also be provided by the OSS promotion forum as a recommended industrial masking sheet. The typical form of the masking sheet contains the list of unit codes with check boxes. The checked units are meant to be one of the necessary elementary skills for the specific job.

On the other hand, the prospective employee prepares his personal record sheet whose form is similar to the masking sheet. The personal record sheet is typically issued from education and training bodies, or can be issued from authorized certificate agencies which administrate the testing procedures. An attempt is made to match the personal record sheet and industrial masking sheet in the course of the employment interview. In some cases, the prospective employee finds some units

are missed as shown in Figure 5.

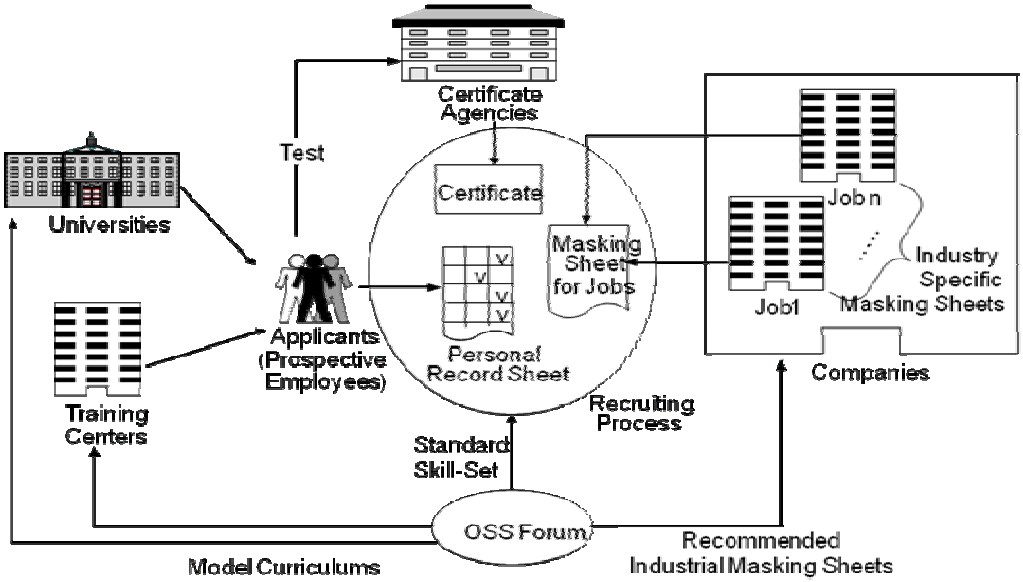

Figure 4. Usage flow of Industrial Masking Sheet and Personal Record Sheet

Personal Record Sheet from prospective employee

Category(9)	Skills(37)	Topic Check Boxes
Basic	Knowledge of OSS	(V) 1-1-1-I ( ) 1-1-1-II
		(V) 1-1-2-II (V) 1-1-2-III
		( ) 1-1-3-II
	...	

Mismatch

Category(9)	Skills(37)	Topic Check Boxes
Basic	Knowledge of OSS	(V) 1-1-1-I ( ) 1-1-1-I
		(V) 1-1-2-II (V) 1-1-2-III
		(V) 1-1-3-II
	...	

Industrial Masking Sheet presented by a company

Figure 5. Example of matching procedure between Industrial Masking Sheet and Personal Record Sheet

2. Detailed Description of Skill-sets

This section provides detailed descriptions of skill-sets that are defined for model curriculum for embedded application developers. The curriculum includes 9 categories and 35 skills as shown in Table 1. Shaded skills in the table are covered in this section.

Table 1 Category of OSS knowledge and skill-sets

Category	WG2 OSS Skills	Code
Basic	Knowledge of OSS	1-1
	Legal Affairs	1-2
	Computer System and Architecture	1-3
	Distributed Architecture	1-4
System	Concept of Linux and Basic Operations	2-1
	Kernel of Linux	2-2
	Linux System Management	2-3
	Linux System Programming	2-4
	Network Server Management	2-5
	Cluster System Architecture	2-6
	Concurrent System Programming	2-7
	Java EE Application Server	2-8
Network	Network Architecture	3-1
	Network Management	3-2
	Network Programming	3-3
Programming	Java	4-1
	C	4-2
	C++	4-3
	Light Weight Language	4-4
	GUI	4-5
	Web Programming	4-6
Multimedia System	Multimedia Service Platform	5-1
	Multimedia Programming	5-2
Development System	Development Frameworks	6-1
	Development Tools	6-2
	Integrated Development Environment	6-3
Security	Encryption	7-1
	Network Security	7-2
	OS Security	7-3
RDB	Basic Skills in RDB	8-1
	RDB System Management	8-2
	RDB Application Development	8-3
Embedded Software	Embedded System Development	9-1
	Embedded Development Environment	9-2
	Device Driver Development	9-3

2.1 Knowledge of OSS

<table>
<tr><td>SKILL CATEGORY NAME</td><td colspan="2">Basic Knowledge</td><td>SKILL CATEGORY NO.</td><td>1</td></tr>
<tr><td>SKILL NAME</td><td colspan="2">Knowledge of OSS</td><td>SKILL NO.</td><td>1</td></tr>
<tr><td>TOPICS</td><td>LEVEL</td><td colspan="2">SUBTOPICS</td><td>CODE NO.</td></tr>
<tr><td rowspan="5">Introduction to OSS</td><td rowspan="4">I</td><td>Objective</td><td colspan="2">Understand the definition of open source software (OSS) and the concept of OSS, OSS development models.</td><td rowspan="4">1-1-1-I</td></tr>
<tr><td>Prerequisite</td><td colspan="2"></td></tr>
<tr><td colspan="3">- Understanding the definitions of Open Source Software (OSS)
- Understanding the concepts of OSS
- Understanding the reasons for business focus on OSS
- Understanding OSS development models
- Understanding introduction to basic OSS licenses</td></tr>
<tr><td rowspan="3">II</td><td>Objective</td><td colspan="2">Understanding situations on deployment and development of OSS</td><td rowspan="3">1-1-1-II</td></tr>
<tr><td>Prerequisite</td><td colspan="2"></td></tr>
<tr><td colspan="3">- Understanding OSS business models
- Understanding type of OSS communities
- Understanding difference in concept between Free Software and OSS</td></tr>
</table>

<table>
<tr><td rowspan="3">History of UNIX and Linux</td><td rowspan="3">I</td><td>Objective</td><td>Describe the history of UNIX from its start in the late 1960s, development into BSD, and contribution to the appearance and development of Linux. In connection with these topics, also address GNU projects and the Free Software Foundation (FSF).</td><td rowspan="3">1-1-2-I</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">- Understanding the history of UNIX
- Understanding the history of Linux
- Understanding the history of GNU Projects</td></tr>
<tr><td rowspan="3">OSS servers</td><td rowspan="3">I</td><td>Objective</td><td>Understanding that OSS is used most widely in Internet servers (mail, Web, and DNS) and a variety of network servers (file sharing and application servers). Also being aware that OSS middleware is gaining more popularity.</td><td rowspan="3">1-1-3-I</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">- Knowing OSS server software
• Mail servers
• Web servers
• DNS servers
• File servers
• Application servers
- Knowing OSS middlewares
- Knowing OSS virtualization tools</td></tr>
</table>

<table>
<tr><td rowspan="3">OSS development tools</td><td rowspan="3">I</td><td>Objective</td><td>Introduce C/C++, Java, PHP, Perl, Python, and Ruby as languages for OSS development. Stress that these languages are developed in turn using OSS. Explain OSS development frameworks for PHP, Ruby, and Java. Also give a brief description of Eclipse, NetBeans, and WideStudio as OSS integrated development tools.</td><td rowspan="3">1-1-4-I</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">Knowing C/C++ compilers
Knowing Java environment
Knowing Java tool chain
Knowing PHP environment
Knowing PHP toolkits
Knowing Perl and CPAN
Knowing Python
Knowing Ruby and Ruby on Rails
Knowing development frameworks for PHP, Ruby, and Java
Knowing integrated development tools
Eclipse
NetBeans
WideStudio</td></tr>
<tr><td rowspan="2">OSS desktop applications</td><td rowspan="2">I</td><td>Objective</td><td>Introduce window systems (integrated desktop environment), office suites, browsers, mailers, and graphic tools as desktop applications based on OSS.</td><td rowspan="2">1-1-5-I</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
</table>

		– Knowing window systems • X window systems • Window Manager • 3D effected window – Knowing integrated desktop environment • KDE • GNOME • XFCE4 – Knowing office suites • OpenOffice.org • Koffice – Knowing browsers • Mozilla firefox • Ephineny • Galeon – Knowing mail clients and schedulers • Mozilla Thunderbird • Evolution • Sylpheed		
OSS standardization	II	Objective	Outline the trend of OSS standardization. Introduce Linux Standard Base (LSB), which is a standard for Linux, and Java standard specifications to clarify the significance of such standardization. Also address efforts in Asian regions toward standardization and relations with international standardization organizations.	1-1-6-II
		Prerequisite		

		– Understanding trends of OSS standardization – Understanding Linux Standard Base (LSB) – Understanding Java standard specifications – Understanding NEAOSS Standardization		
OSS server applications	I	Objective	Introduce OSS-based server applications – Customer Relation Management (CRM), Business Integrated Data Analysis (BI), Enterprise Resource Planning (ERP), and Content Management System (CMS) for Websites. Also introduce other OSS applications for specific business tasks.	1-1-7-I
		Prerequisite		
		– Knowing ERP application ■OpenERP -Knowing CMS application ■Zope ■Plone -Knowing CRM application ■SugerCRM -Knowing BI application		
Use of OSS	I	Objective	Outline the current use of OSS by businesses, using data from various studies. Describe the market shares of Linux and OSS middleware. Also introduce opinions of business users concerning their interest in adopting OSS and the advantages and disadvantages they perceive in OSS.	1-1-8-I
		Prerequisite		

<table>
<tr><td></td><td></td><td colspan="2">-Understanding Current use of OSS by businesses
■The market shares of Linux and OSS middleware
■The advantages and disadvantages
-Understanding Web system construction
■How OSS is used and points to be clarified in system construction
■Pros-and-cons of OSS
■Precautions to be taken in system construction</td><td></td></tr>
<tr><td rowspan="3">Web system development</td><td rowspan="3">I</td><td>Objective</td><td>Using a typical example of Web system construction, explain how OSS is used and clarify points in system construction, pros-and-cons of OSS, and precautions to be taken in system construction.</td><td rowspan="3">1-1-9-I</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">• Understanding typical web system stacks
• Understanding pros and cons of OSS on Web system construction</td></tr>
<tr><td rowspan="3">OSS communities</td><td rowspan="3">I</td><td>Objective</td><td>Giving a typical example of an OSS community, outline the types and features of OSS communities. Also explain how to join such a community and what precautions to take in participating in one.</td><td rowspan="3">1-1-10-I</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">● Understanding types and features of OSS communities and participation in them
● Understanding communities’ enabler services
● **** Searching development project
● Understanding how to join a project</td></tr>
<tr><td rowspan="2">OSS business</td><td rowspan="2">II</td><td>Objective</td><td></td><td rowspan="2">1-1-10-II</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
</table>

		• Understanding business model • Package business • Customizing OSS • Dual licensing • Subscription models • Installation service • Professional services • Support business • Consulting business • Education business		
Earning OSS information	II	Objective		1-1-11-II
		Prerequisite		
		-Project site -sourceforge -Freshmeat -Information sites provided by non profit organizations -News site -Community site		
Deployment of OSS operating system	II	Objective	Understanding basics of OSS operating system.	1-1-12-II
		Prerequisite		

<table>
<tr><td></td><td></td><td colspan="2">-Supported hardware and drivers
-OSS Applications
-Enterprise Linux distributions
■SuSE Linux Enterprise Server (SLES)
■Red Hat Enterprise Linux (RHEL)
-Community based Linux distributions
■KNOPPIX
■Ubuntu
-OS other than Linux
■FreeBSD
■NetBSD</td><td></td></tr>
<tr><td rowspan="3">Deploying server application</td><td rowspan="3">II</td><td>Objective</td><td>Understanding the existence of several kinds of OSS server applications.</td><td rowspan="3">1-1-13-II</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">-MTA
■sendmail
■Postfix
-HTTP Server
■Apache HTTP Server
-Installation from source code
■Configure and create
-Installation by binary package
■Debian package
■RPM package form</td></tr>
<tr><td rowspan="2">Deployment of OSS server</td><td rowspan="2">II</td><td>Objective</td><td>Understanding how to install several OSS server applications.</td><td rowspan="2">1-1-14-II</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
</table>

		-Installation of database management server ■PostgreSQL ■MySQL -Details of MySQL installation ■Installation of business edition ■Installation of community edition -Installation of Network and Server management software ■OpenNMS ■Hinemos		
Deployment of OSS desktop application	II	Objective	Understanding how to install and configure several OSS desktop applications.	1-1-15-II
		Prerequisite		

		-Browser ■Mozilla firefox and its configuration ■Extensions -Mail client ■Mozilla thunderbird and its configuration ■Connection to POP server ■Connection to IMAP4 server -Mail and schedule client ■Evolution and its configuration ■Connection to MS exchange server ■Connection to IMAP server ■Connection to POP3 server -Office applications ■OpenOffice.org ■Gimp -OpenClient		
Deployment of OSS server application	II	Objective		1-1-16-II
		Prerequisite		
		-CMS/blogs/SNS applications ■XOOPS cube ■wordpress -SNS application such as OpenPNE, SKIP		
Deployment of virtualization software	II	Objective		1-1-17-II
		Prerequisite		

		-Introduction to virtualization ■Host OS virtualization ■Hypervisor virtualization -Concept of Xen ■Domain 0 ■Domain U -Installation of Xen ■Installation of guest OS -Configuration of Xen	

2.2 Kernel of Linux

SKILL CATEGORY NAME	System			SKILL CATEGORY NO.	2
SKILL NAME	Kernel of Linux			SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS			CODE NO.
A general introduction to the kernel of Linux	I	Objective	Understanding the basic roles of OS and the history of OSes which have been developed to date. Knowing today's most popular OSes. Understanding features of each OS.		2-2-1-I
		Prerequisite	None		

		- Understanding the basic roles of OS - Understanding the background to the deployment of an OS on the computers - Understanding the history of OSes - Understanding introduction to today's popular OSes - Understanding the type and features of OSes - Understanding the basics of system call - Understanding the basics of context switch		
Architecture of Linux kernel	I	Objective	Understanding the basic architecture and functions of the Linux kernel, including process management, memory management, file systems, networking, and I/O processing.	2-2-2-I
		Prerequisite	None	

		- The architectural view of Linux kernel - Understanding basics of process management - Understanding basics of memory management - Understanding basics of file systems - Understanding basics of networking - Understanding basics of I/O processing		
Scheduling	I	Objective	Understanding the concept of context switch, and outlining the mechanism of switching processes. Knowing the actual application of scheduling algorithms, in addition to the underlying queuing theory and Markov chain.	2-2-3-I

<table>
<tr><td rowspan="2"></td><td rowspan="2"></td><td>Prerequisite</td><td></td><td rowspan="2"></td></tr>
<tr><td colspan="2">- Understanding the concept of context switch
- Understanding the mechanism of switching processes
- Understanding the concept of scheduling algorithm
task list queuing
• Real time scheduling
• Dual running queue
- Using I/O scheduling
• CFQ
• Dead timeout
• Elevator algorithm
- Understanding the selection and difference of kernel timers
• TSC
• ACPI timer
• High resolution timer, HPET</td></tr>
<tr><td rowspan="2">Interrupts and delayed action</td><td rowspan="2">I</td><td>Objective</td><td>Understanding the association between the CPU and other devices and the concept of “interrupt.” Knowing several topics on interrupt, including those of the management and types of interrupts. Also understanding the outline of interrupt processing in the kernel and the basic idea of the time-sharing system.</td><td rowspan="2">2-2-4-I</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
</table>

<table>
<tr><td></td><td></td><td colspan="2">• Understanding the basics of interrupts
■ Hardware interrupt
■ Software interrupt
■ Timer interrupt
• Understanding the concept of delayed execution
■Work queue
■Wait queue
■</td><td></td></tr>
<tr><td rowspan="3">System calls</td><td rowspan="3">I</td><td>Objective</td><td>Understanding the system call function used to use OS functions in ordinary applications. Understanding the position of the system call among other functions. Knowing the outline of the internal behavior of the OS when the system call function is activated.</td><td rowspan="3">2-2-5-I</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">• Understanding concept of system call
• Understanding the trap mechanism
• Understanding switch of address space
■ Kernel space
■ User space
■ Copying data from/to kernel and user space
• Understanding the basic method for implementing features of system call
■ Dispatch table</td></tr>
<tr><td rowspan="2">Process management</td><td rowspan="2">I</td><td>Objective</td><td>Understanding the lifecycle of a process as it is created, is completed, and then disappears. Introduce a series of topics related to the lifecycle of processes, including the internal data structure of a process, state transition, process groups, and process creation methods.</td><td rowspan="2">2-2-6-I</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
</table>

<table>
<tr><td></td><td></td><td colspan="2">-Understanding the concept of processes in multitask OS
-Understanding process lifecycle and its process
-Understanding the concept of process space
-Understanding the transition of running modes
-Understanding the concept of process context and kernel context
-Understanding the creation of a process and thread
■fork()/clone()
■wait()</td><td></td></tr>
<tr><td></td><td>II</td><td>Objective</td><td>Understanding deadlock and other situations where processes need synchronization or exclusion, as well as management by the kernel of such situations. Knowing mounting technology to realize exclusive control, in addition to basic theories underlying the technologies.</td><td>2-2-6-II</td></tr>
<tr><td></td><td></td><td>Prerequisite</td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="2">-Understanding the synchronization and exclusive control
-Understanding bad situations such as deadlock
-Understanding the control method of deadlocks in kernel
-Understanding mounting technology for exclusive control</td><td></td></tr>
<tr><td>Memory management</td><td>I</td><td>Objective</td><td>Understanding the concept of “process space,” in which a program runs, and the purpose and features of process spaces. Knowing what can be done in one space but cannot be done in another. Also understanding the transition of running modes caused by a system call.</td><td>2-2-7-I</td></tr>
<tr><td></td><td></td><td>Prerequisite</td><td></td><td></td></tr>
</table>

		-Understanding the principle of physical address management ■Page ■Buddy system ■Grab and release of page -Understanding the algorithm of dynamic page allocation ■Slab allocator ■Memory pool ■Algorithms for assigning memory -Understanding the concept of swap and paging ■Demand paging ■Page cache ■Page fault ■Dirty page	

<table>
<tr><td rowspan="4">File Systems</td><td rowspan="3">I</td><td>Objective</td><td>Understanding the concept of the virtual filesystem layer which is an abstract layer for hiding differences in the filesystem in order to access seamlessly. Additionally understanding file operation in the kernel and the concept of special files.</td><td rowspan="3">2-2-8-I</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">-Understanding virtual file systems
■Inode
■Directory
■Mount
-Understanding file operation
■Open/close
■Seek
■Read/write
-Understanding special files
■Device file
■Proc filesystem
■Sys filesystem</td></tr>
<tr><td rowspan="3">II</td><td>Objective</td><td>Understanding the concept of asynchronous access to file and devices to archive high performance.</td><td rowspan="3">2-2-8-II</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">• Understanding asynchronous file operations
■ AIO subsystem
• Understanding the concept of direct device access
■ Direct IO</td></tr>
</table>

<table>
<tr><td rowspan="3">Inter process communications</td><td rowspan="3">I</td><td>Objective</td><td>Understanding shared memory, semaphore, messaging, and other functions provided as a means of communicating data and messages between processes.</td><td rowspan="3">2-2-9-I</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">Understanding IPC Interprocess communication
Shared memory
Semaphore
Messaging</td></tr>
<tr><td rowspan="2">Protocol stacks</td><td rowspan="2"></td><td>Objective</td><td>Understanding the concept of OSI reference models and TCP/IP protocol stacks. Knowing TCP/IP as an example of practical deployed protocol. Understanding the concept of IP address, TCP port number and other advanced features provided by TCP/IP protocol stack.</td><td rowspan="2">2-2-10-I</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
</table>

		• Understanding socket interfaces ▪ UNIX domain socket ▪ INET socket • Understanding IP and UDP ▪ IP address ▪ IPv4 and IPv6 ▪ Routing • Understanding UDP and TCP -Difference between two major protocols - TCP flow control and congestion control	

2.3 Embedded Systems Development

SKILL CATEGORY NAME	Embedded Software		SKILL CATEGORY NO.	9
SKILL NAME	Embedded Systems Development		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS		CODE NO.

<table>
<tr><td rowspan="4">Basics of Embedded computer system</td><td rowspan="4">I</td><td>Objective</td><td>Understanding the basic concept and substance of embedded computer systems. Describe the general architecture of hardware, software, and networks in embedded computer systems.</td><td rowspan="4">9-1-1-I</td></tr>
<tr><td>Prerequisite</td><td>-</td></tr>
<tr><td colspan="2">- Basics of Embedded computer system
- Application of embedded computer system
- Architectures for embedded computer systems
- Software architecture for embedded systems
- Network architecture for embedded systems
- Design methods for embedded computer systems
- Needs and application for OSS embedded computer systems
- Relations between embedded systems and OSS
- Practical cases of OSS embedded OSS systems
- Precautions to be taken in using OSS in embedded systems</td></tr>
<tr><td rowspan="3">Architecture of Embedded computer system</td><td rowspan="3">I</td><td>Objective</td><td>Understanding basic structure and roles and characteristics of embedded computer systems</td><td rowspan="3">9-1-2-I</td></tr>
<tr><td>Prerequisite</td><td>9-1-1-I</td></tr>
<tr><td colspan="2">- Characteristics of embedded hardware
- Requirement of embedded computer system
- Limitation on hardware, software and network</td></tr>
</table>

<table>
<tr><td rowspan="4">Requirements for embedded computers</td><td rowspan="4">I</td><td>Objective</td><td>Understanding the basic configuration and features of embedded computers. Explain that connectivity, reliability, efficiency and interchangeability are among the requirements for computer systems.</td><td rowspan="4">9-1-3-I</td></tr>
<tr><td>Prerequisite</td><td>9-1-1-I</td></tr>
<tr><td colspan="2">- Basic configuration and features
- Connectivity, reliability, efficiency and interchangeability
-</td></tr>
<tr><td rowspan="4">Hardwares and architectures for embedded computers</td><td rowspan="4">I</td><td>Objective</td><td>Understanding characteristics of typical hardwares and architectures for embedded software</td><td rowspan="4">9-1-4-I</td></tr>
<tr><td>Prerequisite</td><td>9-1-3-I</td></tr>
<tr><td colspan="2">- Typical CPUs
■ Such as ARM, MIPS, Xscale, PowerPC, SH, M32R, etc.
- Typical OS
■ Such as uCLinux, T-Kernel, eCOS, RTLinux, etc.
- Typical network connectivity
■ LAN, WiFi, Infrared, low bandwidth wireless, IEEE1394, etc.
- Issues and countermeasures specific to embedded systems
■ Power consumption
■ Small memory
- Basic hardware configuration of embedded systems
■ The basic configuration, roles, and features of embedded computer hardware.
-</td></tr>
<tr><td>Embedded hardware</td><td>II</td><td>Objective</td><td></td><td></td></tr>
</table>

<table>
<tr><td>architecture</td><td></td><td>Prerequisite</td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="2">- Architecture of CPUs
■ Internal CPU architecture
■ System buses
■ Bus data communication
■ Operations in the CPU.
■ Memory connections
■ Instruction Set Architecture
- Memory management methods
■ Flash memory management
■</td><td></td></tr>
<tr><td rowspan="3">Embedded software basics</td><td rowspan="3">I</td><td>Objective</td><td>Understanding roles and features of embedded software. Such as task and scheduling as a basic part of embedded software processing.</td><td rowspan="3">Code shared with Linux kernel skill-sets</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">- Functions of embedded OS kernels
■ Task and scheduling
■ Preemption
- Resource management
■ Physical memory addressing
■ Virtual memory addressing</td></tr>
<tr><td>Details of embedded OS kernels</td><td>II</td><td>Objective</td><td>Understanding functions, characteristics and roles of embedded OS kernels in detail.</td><td>Only embedded specific items

9-1-6-II</td></tr>
</table>

		Prerequisite		
		- Functions required of embedded OS kernels ■ Embedding-specific issues related to task control ■ Synchronization primitives - Implementation of synchronization ■ Test and set ■ Semaphores		
Real time system	I	Objective	Understanding necessity, characteristics and processing method and design of real time systems.	9-1-7-I
		Prerequisite		

<table>
<tr><td rowspan="4">Real time system</td><td></td><td colspan="2">- Basics of real time system
- Design patterns for RT system
- Time management
- Real time programming
- Queuing techniques
- Hard realtime
- Soft realtime
- Device driver and RT
- Priority and exclusion order</td><td></td></tr>
<tr><td rowspan="3">II</td><td>Objective</td><td>Understanding the mechanism of the context switch used to perform multiple tasks, and the concept of asynchronous processing and task priority control.</td><td rowspan="3">9-1-7-II</td></tr>
<tr><td>Prerequisite</td><td>2-4-5-I (process switching), 2-4-6-II (interrupt)</td></tr>
<tr><td colspan="2">- Application demand context switch
  - Preemption-based implementation
  - Interrupt-based context switch
- Application controlled asynchronous / synchronous processing
  - Execution modes
  - Interrupt priority
- Task priority control
- Time control methods inside Linux kernel</td></tr>
</table>

I/O Resource management	II	Objective	Understanding concepts of system resource allocation, sharing and management in embedded environment including memory and IO resources.	9-1-8-II
		Prerequisite	2-2-*-II (memory mgmt of Linux), 2-4-*-II (semaphore)	
		- Management bus such as I2C, SMB, smart battery - I/O connection and buses such as Universal Serial Bus, IEEE1394, GPIB etc. - Special devices such as sensors, camera, motor and actuators - Communication devices such as WCDMA, GSM, bluetooth - User Interface devices such as touch panel, keypad, 3D accelerometer, thermometer etc.		
- Special memory devices		Objectives	Understanding characteristics of memory devices specific to embedded systems.	9-1-9-II

management		Prerequisite	2-2-*-II (memory mgmt of Linux)	
		■ Flash memory ■ SSD ■ Memory cards such as SD card, MMC, memory stick, etc.		
High reliability implementation	II	Objective	Understanding perspective and technique of improving reliability of embedded application.	Move to appendix 9-1-10-II
		Prerequisite	-	
		- High reliability implementation - Fault avoidance and fault tolerance - Error detection implementations - Error recovery implementations		

2.4 Embedded Development Environment

<table>
<tr><td>SKILL CATEGORY NAME</td><td colspan="3">Embedded SW</td><td>SKILL CATEGORY NO.</td><td>9</td></tr>
<tr><td>SKILL NAME</td><td colspan="3">Embedded Development Environment</td><td>SKILL NO.</td><td>2</td></tr>
<tr><td>TOPICS</td><td>LEVEL</td><td colspan="3">DESCRIPTIONS & SUBTOPICS</td><td>Topic Code</td></tr>
<tr><td rowspan="5">Building Projects</td><td>I</td><td colspan="3">-</td><td>-</td></tr>
<tr><td rowspan="4">II</td><td>Objective</td><td colspan="2">To have ability to build own projects by fluently writing Makefiles that can build and install static and/or dynamic libraries.</td><td rowspan="4">9-2-1-II
(OR code share with development system-development-?-?6-2-?-II)</td></tr>
<tr><td>Prerequisite</td><td colspan="2"><using GCC/G++></td></tr>
<tr><td colspan="3">- Able to write Makefiles
■ Primitives of Makefile (target, dependency, rule)
■ MACRO and .SUFFIXES
■ Pre-defined macro
■ $<, $@, $?
■ Make depend
- Able to write Recursive Makefiles
■ Multiple and hierarchical directories
- Able to write Makefiles for building and installing libraries
■ Static libraries
■ Shared libraries - -fPIC & -shared options in GCC
■ PATH and Environment variables
■ Linux commands – ar, ldconfig</td></tr>
<tr><td>III</td><td colspan="3">-</td><td>-</td></tr>
<tr><td rowspan="2">Managing Project</td><td rowspan="2">I</td><td>Objective</td><td colspan="2">To have ability to use tools for managing OSS projects collaboratively.</td><td rowspan="2">9-2-2-I
(OR code share with development</td></tr>
<tr><td>Prerequisite</td><td colspan="2"></td></tr>
</table>

<table>
<tr><td rowspan="3"></td><td></td><td colspan="2">- Understanding types of version control systems
■ Revision control system (RCS)
■ Concurrent Versioning System (CVS, Subversion)
■ Distributed Versioning System (GIT, BAZZAR)
- Able to Use CVS
■ Rationales for CVS
■ Setting up CVS server
■ Initializing project
■ Using CVS commands (check-in, check-out, etc)
■ Performing Backup and recovery
- Understand the characteristics of other tools
■ Subversion
■ Etc</td><td>system-development-?-?6-2-?-II)</td></tr>
<tr><td>II</td><td colspan="2"></td><td></td></tr>
<tr><td>III</td><td colspan="2"></td><td></td></tr>
<tr><td rowspan="4">GNU Tools</td><td rowspan="3">I</td><td>Objective</td><td>To understand the roles of basic GNU tools and their relationships.</td><td rowspan="3">CODE SHARE
6-2-?-I</td></tr>
<tr><td>Prerequisite</td><td><using GCC/G++></td></tr>
<tr><td colspan="2">- Able to use basic GNU tools for compiler, load, library management
- Understand relationships of GNU tools
■ Roles and relationships in gcc, make, automake, autoconf, //gdb</td></tr>
<tr><td rowspan="2">II</td><td>Objective</td><td>To have ability to use advanced GNU tools for releasing and managing source codes.</td><td rowspan="2">CODE SHARE
6-2-?-II</td></tr>
<tr><td>Prerequisite</td><td>9-2-1-II</td></tr>
</table>

		- Able to use autoconf ■ Rationales for autoconf ■ Configure script ■ Configure.ac ■ Macros - Able to use automake ■ Rationales for automake ■ Makefile.in ■ Makefile.am ■ Recursive Makefile.am - Use more tools – gcov, ctags, cscope		
	III			-
Building Cross Development	I	Objective	To have ability to build cross development environments.	9-2-4-I
		Prerequisite		

<table>
<tr><td rowspan="3">Environment</td><td></td><td colspan="2">- Understand frameworks of cross development environment
■ Target and Host
■ Connections between target & host
- Able to use minicom
■ Connecting serial connection
■ Setting up minicom – baud rate, parity, etc.
- Able to use NFS
■ Understanding NFS
■ Setting NFS between host and target
- Able to use binutils and cross compiler
■ Target compilers - Gas, gcc
■ Lib? and includes for target machine
■ Setting environment variables (INC, LD_LIB, CC, PATH, etc) for cross compiler</td><td></td></tr>
<tr><td>II</td><td colspan="2"></td><td></td></tr>
<tr><td>III</td><td colspan="2">-</td><td></td></tr>
<tr><td rowspan="3">Debugging with gdb</td><td rowspan="3">I</td><td>Objective</td><td>To have ability to debug source code using debugging tools such as gdb.</td><td rowspan="3">CODE SHARE
6-2-?-II</td></tr>
<tr><td>Prerequisite</td><td><using GCC/G++></td></tr>
<tr><td colspan="2">- Able to use debugging tool (gdb)
■ Using Gcc options
■ Using gdb commands – Run, Step, Next, Trace, Break, Watch point, Inspection for variable and register
■ Debugging multithread programs
■ Debugging libraries</td></tr>
</table>

<table>
<tr><td></td><td>II</td><td colspan="2"></td><td></td></tr>
<tr><td></td><td>III</td><td colspan="2"></td><td></td></tr>
<tr><td rowspan="3">Bug Tracking</td><td rowspan="2">I</td><td>Objective</td><td>To have ability to install and use bug tracking tools.</td><td rowspan="2">CODE SHARE
6-2-?-II</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">- Able to use basic operations of bugzilla
- Able to install bugzilla server
- Understand other bug tracking tools
■ Issue tracking – Mantis, Trac</td></tr>
<tr><td>II</td><td colspan="2"></td><td>-</td></tr>
<tr><td>III</td><td colspan="2"></td><td>-</td></tr>
<tr><td rowspan="5">Remote Debugging</td><td rowspan="3">I</td><td>Objective</td><td>To have ability to setup remote debugging environment.</td><td rowspan="3">9-2-7-I</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">- Able to setup remote debugging environment with gdb
■ Concept of remote Gdb protocol
■ Running gdbserver on target machine
■ Attaching
■ Command-line arguments for gdbserver
- Able to use remote debugging environment with gdb
■ Connecting to gdbserver
■ Sending files to remote target
■ Monitor command to gdbserver
■ Remote configuration</td></tr>
<tr><td>II</td><td colspan="2"></td><td></td></tr>
<tr><td>III</td><td colspan="2"></td><td></td></tr>
</table>

2.5 Device Driver Development

SKILL CATEGORY NAME	Embedded Software			SKILL CATEGORY NO.	9
SKILL NAME	Device Driver Development			SKILL NO.	5
TOPICS	LEVEL	DESCRIPTIONS & SUBTOPICS			Topic Code
Linux Kernel	I	Objective	Understand the basic Linux Architecture.		CODE SHARE 2-2-1-I
	II	Objective	Use kernel primitive functions.		9-5-1-II
		Prerequisite			
		- Use various kernel memory primitives including kmalloc() / kfree(), vmalloc() / vfree(), alloc_page() / free_page(), slab primitives - Understand the process control primitives such as schedule(), sleep() and wakeup() - Understand kobject - Use primitive functions for /proc, /sys file system - Use the macros and primitives to handle kernel lists and queues			
	III	-			-
Kernel Debugging	I	Objective	Setup remote debugging environment and debug.		CODE SHARE 9-2-7-I
	II	Objective	Monitor and debug kernel remotely.		9-5-2-II
		Prerequisite			

		- Monitor kernel event and behavior with /proc, /sys, prink() - Use kgdb to trace kernel code, watch kernel data - Debug LKM (loadable kernel module)		
	III			-
Kernel Configuration	I	Objective	Configure Linux Kernel.	9-5-3-I
		Prerequisite		
		- Configure Linux Kernel with kconfig system - Build kernel image (vmlinux, bzImage) both in source directory and separate object directory - Configure, build and install kernel module - Configure Linux kernel for remote debugging with kgdb (serial or Ethernet) - Configure Linux kernel for detailed monitoring with /proc, /sys file system and		
	II	Objective	Understand the method to add new items to kernel.	9-5-3-II
		Prerequisite		
		- Understand the method to add new items to kernel - Modify kconfig system		
	III	-		
Character Device Driver	I	-		
	II	Objective	Implement character device driver as a kernel module.	9-5-4-II
		Prerequisite		
		- Register and unregister character device driver with specified major, minor number - Implement system call API such as open(), read(), write(), ioctl(), release() - Implement data availability API such as poll(), fasync()		
	III	Objective	Implement device driver which uses interrupt and/or DMA.	9-5-4-III

		Prerequisite		
		- Implement device drive which uses hardware interrupt by setting up IRQ - Understand execution context (user / kernel / interrupt) - Understand do's and don'ts in ISR - Understand and use delayed execution such as Top Half/Bottom Half		
Other Device Drivers	I	-		-
	II	-		-
	III	Objective	Implement all kinds of device drivers.	9-5-6-III
		Prerequisite		
		- Implement the bus drivers including I2C, SPI, PCI, PCMCIA/CF - Understand USB software architecture and Implement USB class driver - Implement block device driver - Implement network interface driver - Implement video frame Buffer Driver - Implement audio driver using open sound system - Implement camera driver for video for Linux		
Building Target System	I	Objective	Configure boot loaders and manage boot sequences, Build root file system.	9-5-7-I
		Prerequisite	SYSTEM-LINUX SYSTEM MANAGEMENT (2-3-?-I)	

<table>
<tr><td rowspan="3"></td><td></td><td colspan="2">- Configure bootloader such as uBoot, Redboot, and etc.
- Use flash commands to fuse bootloader/kernel/root file system
- Configure DHCP/BOOTP/NFS for network boot
- Use boot load command or specify kernel options for network boot
- Build root file system from scratch
- Build Initramfs file system
- Configure, build and install busybox</td><td></td></tr>
<tr><td>II</td><td colspan="2">-</td><td></td></tr>
<tr><td>III</td><td colspan="2">-</td><td></td></tr>
<tr><td rowspan="3">Kernel Synchronization Primitives</td><td>I</td><td colspan="2">-</td><td></td></tr>
<tr><td rowspan="3">II</td><td>Objective</td><td>Implement device driver with concurrent access.</td><td rowspan="3">9-5-9-II</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">- Implement process level synchronization using sleep() / wakeup() / wait_event()
- Implement mutual exclusion with kernel mutex primitives such as spin lock, read write lock, atomic operations
- Understand when spin_lock_irq() is required</td></tr>
<tr><td>III</td><td colspan="2">-</td><td></td></tr>
<tr><td rowspan="3">Kernel Thread</td><td>I</td><td colspan="2"></td><td>(code share)
2-2-6-I</td></tr>
<tr><td rowspan="3">II</td><td>Objective</td><td>Implement kernel thread.</td><td rowspan="3">9-5-10-II</td></tr>
<tr><td>Prerequisite</td><td></td></tr>
<tr><td colspan="2">- Create and daemonize a kernel thread
- Understand and use softirq/tasklet</td></tr>
<tr><td>III</td><td colspan="2">-</td><td></td></tr>
<tr><td rowspan="2">Power Management</td><td>I</td><td colspan="2">-</td><td></td></tr>
<tr><td>II</td><td colspan="2">-</td><td></td></tr>
</table>

	III	Objective	Implement power efficient device driver.	9-5-11-III
		Prerequisite		
		- Understand Linux Power Management architecture ■ ACPI states - Understand standby and wakeup mechanism - Use ACPI to manage power consumption		

Blank Page(Don’t remove)

3. Model Curriculum for OSS based Embedded SW Developer

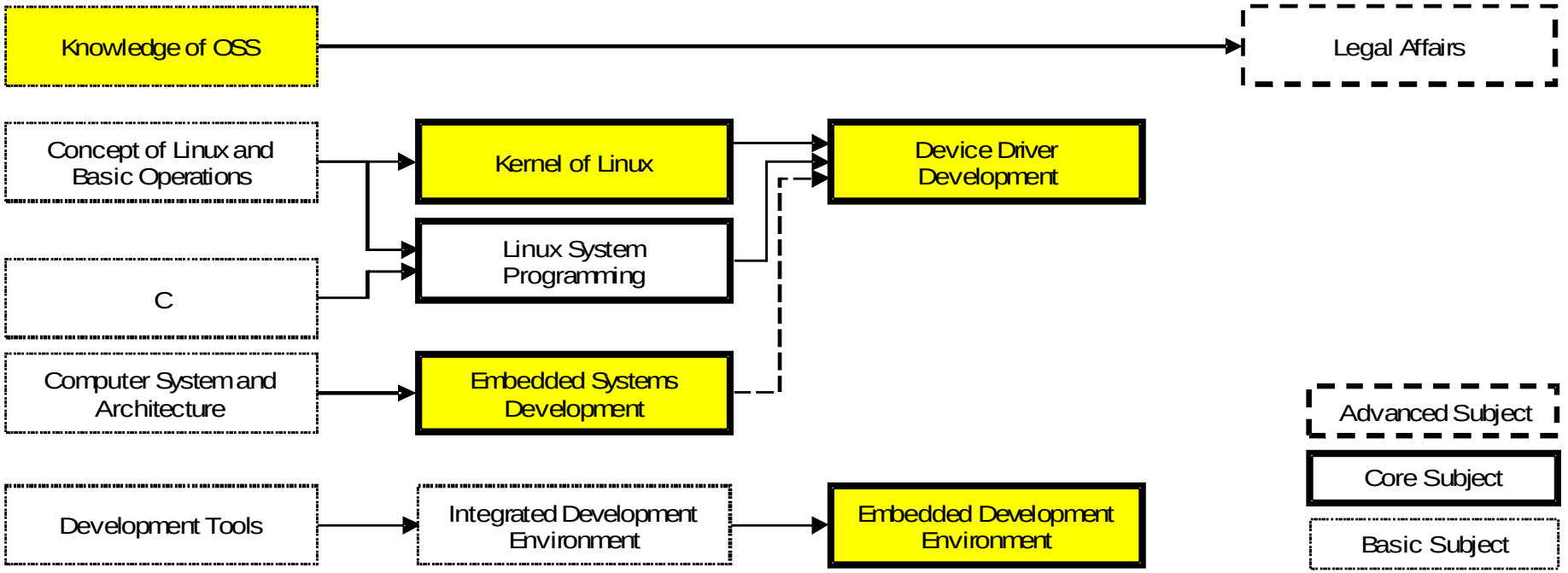

Figure 6. Block diagram of the model curriculum for OSS based embedded SW developer

Figure 6 shows the block diagrams which contain 12 subjects: 5 basic subjects, 5 core subjects and 1 advanced subject. Each subject is composed of units, i.e., the set of leveled subtopics in a same topic. For example, the core subject is a set of units each of which is considered crucial for this curriculum. Figure 7 shows selections of units from corresponding skills for each subject. For example, the subject “Kernel of Linux” consists of units such as Introduction (2-2-1-I), Scheduling (2-2-2-I), Interrupt (2-2-4-I), System Calls (2-2-5-I), Process Management (2-2-6-*), Memory Management (2-2-7-1) and File Systems (2-2-8-I) as one of the core subjects.

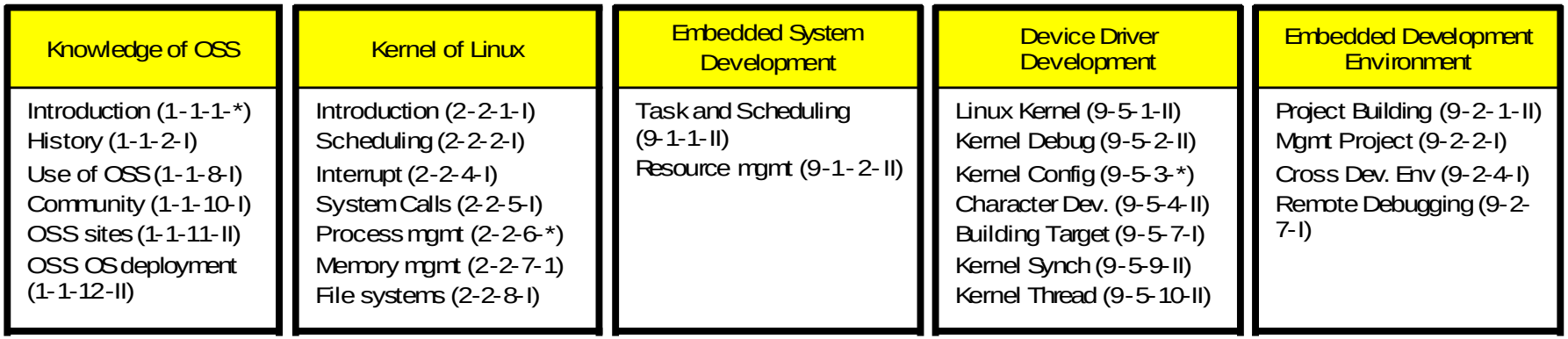

Figure 7. Selection of units for each subject

Appendix (informative)

This appendix provides the skill-sets not covered in Chapter 2. The purpose of providing this appendix is to show our on-going efforts toward the full coverage of 37 skill-set definitions. Thus, please treat this informative part with caution because it may be changed without any notification in the future.

Category	WG2 OSS Skills	Code
Basic	Knowledge of OSS	1-1
	Legal Affairs	1-2
	Computer System and Architecture	1-3
	Distributed Architecture	1-4
System	Concept of Linux and Basic Operations	2-1
	Kernel of Linux	2-2
	Linux System Management	2-3
	Linux System Programming	2-4
	Network Server Management	2-5
	Cluster System Architecture	2-6
	Concurrent System Programming	2-7
	Java EE Application Server	2-8
Network	Network Architecture	3-1
	Network Management	3-2
	Network Programming	3-3
Programming	Java	4-1
	C	4-2
	C++	4-3
	Light Weight Language	4-4
	GUI	4-5
	Web Programming	4-6
Multimedia System	Multimedia Service Platform	5-1
	Multimedia Programming	5-2
Development system	Development Frameworks	6-1
	Development Tools	6-2
	Integrated Development Environment	6-3

Security	Encryption	7-1
	Network Security	7-2
	OS Security	7-3
RDB	Basic Skills in RDB	8-1
	RDB System Management	8-2
	RDB Application Development	8-3
Embedded Software	Embedded System Development	9-1
	Embedded Development Environment	9-2
	Device Driver Development	9-3

* Shaded Skills are covered in Section 2

SKILL CATEGORY NAME	Basic		SKILL CATEGORY NO.	1
SKILL NAME	Legal Affairs		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Meaning of OSS licenses	I	(1) Clarify the legal position of OSS licenses (license agreements). (2) Explain how effective licenses are and what licenses mean to such parties as providers, users, and distributors.		1-2-1-I
Selection of OSS license	I	(1) Show the features of typical OSS licenses (GPL, BSDL, MPL). (2) Explain the difference between open source software licenses and commercial licenses.		1-2-2-I
	II	(1) Usage of multiple licenses on same product.		1-2-2-II
Explanation of typical open source licenses [GPL type]	I	(1) Copyright and copyleft. (2) Clarify the relation between software copyright and license. Also explain the background to the appearance of the copyleft. Address the relation between the basic concept of the copyleft and GPL, which embodies the concept. (3) Feature of GPL and the significance of GPL v3 rev. (4) Describe the definition and history of GNU General Public License (GPL). Also explain the points and significance of GPL v3, which was revised in 2007. (5) Various problems concerning GPL. (6) Show GPL has spread so forcibly that GPL is said to be causing “GPL contamination” and that it is difficult for GPL to coexist with software otherwise licensed. Present actual cases of GPL infringement.		1-2-3-I
Explanation of typical open source licenses [MPL type]	I	(1) Features of MPL, and MPL and software patents. (2) Clarify the features of Mozilla Public License (MPL). Explain MPL employs the bazaar system, which characterizes OSS, and that considerations were given to its coexistence with commercial software. Also point out that MPL is more closely related to patents.		1-2-4-I
Explanation of typical open source licenses [BSD type]	I	(1) Advantages and disadvantages of BSDL and Apache licenses. (2) Explain the features of BSD License and Apache Software License, and their pros-and-cons resulting from their least restriction. Introduce the amended BSD		1-2-5-I

		License, referring to well-known “advertisement provision” under the BSD License.	
The outline of intellectual property	I	(1) Copyright of software. (2) The basic idea of copyright as an intellectual property right. (3) The fundamental idea of the relations of copyrights with the software industry. (4) the definition and substance of copyright.	1-2-6-I
	II	(1) Problems and trends concerning software patents. (2) The basic idea of patent as an intellectual property right. (3) The definition of patent, patent systems, and their present status in various countries. (4) Software patents and innovation.	1-2-6-II
	III	(1) Other intellectual property related to software. (2) Trademark. (3) Model utility right. (4) Design right. (5) Trends on the software industry besides copyright and patent. (6) Introduce topics about these rights, including design right infringement.	1-2-6-III
Points to be considered from the perspective of intellectual property at the time of using OSS	II		
Points to be considered from a legal perspective other than intellectual property at the time of using OSS	II		
Legal risk management by corporations/organizations etc.	II		

Businesses prepared for legal risks	III		
Legal risk reduction measures to be considered by OSS development communities	III		
Legal risk reduction measures to be considered by OSS-business related corporations	III		
Examples of lawsuits/problems in connection with OSS intellectual property issues	III		
Controversy over patents for software	III		
Guidelines for patent applications associated with intellectual property	III		

SKILL CATEGORY NAME	Basic		SKILL CATEGORY NO.	1
SKILL NAME	Computer System and Architecture		SKILL NO.	3
TOPICS	LEVEL	SUBTOPICS		CODE NO.
The basics of computer architecture	I	(1) System architecture. (2) Present the basic idea of the substance of a system and the concepts of a computer system. Show that there are various types of system architectures. Explain the substance of system architecture and the outline of system architecture designing.		
The basics of computer hardware	I	(1) Components of the computer system. (2) The basic configuration of the computer system. (3) The role and features of each component. (4) The history of development of the computer system. (5) Relation to OSS. (6) OSS compatibility with device drivers. (7) Open BIOS. (8) The disclosure of technical information of hardware.		
	II	(1) Evaluation indicators for the computer system. (2) The performance of computer hardware. (3) Connectivity (plug-and-play, device driver). (4) Reliability (failure rate, MTTR, MTBF). (5) Efficiency (performance and throughput, performance evaluation). (6) Compatibility (standardization of interfaces, etc.).		

The basics of CPU/memory architecture	I	(1) Basic architecture and role of CPU. (2) The basic architecture, features of CPU and its history. (3) Introduce RISC and CISC as typical architectures. (4) Some types of pipeline processing and multiprocessor architectures as a mechanism to achieve faster operation. (5) Memory system. (6) Memory configuration and storage hierarchy. (7) The basic architecture and role of memory chips and their history. (8) Introduce the role of a memory chip used as a main memory unit, the hierarchic storage system composed of primary and secondary caches, and methods of connecting memory devices with the CPU. (9) The role of each memory unit in the computer.	
Basic architecture and roles of peripheral devices	I	(1) Various types of peripheral devices. (2) Explain their basic architecture, role, feature, and history. (3) FD, MO, and CD-R as auxiliary storage devices. (4) Keyboard, mouse, scanner, tablet as input devices. (5) Display and printer as output devices.	
The basics of interface technology	I	(1) Features of various interfaces. (2) The basic architecture, role, and features of interfaces and their history. (3) USB, IEEE1394, and eSATA as hot-pluggable interfaces. (4) SCSI, IDE, PCI, and serial types as legacy interfaces.	

	II	(1) Fiber channel, iSCSI.	
Software architecture		(1) Components of computer software. (2) The operating system (OS), middleware, and applications, explain their function, architecture, and history. (3) Implementation of OSS version of softwares.	
OS architecture		(1) The basic architecture, role, features, and history of OS. (2) Outline of the current status of implementation of OSS and its mechanism. (3) Resource management of files, networks, input-output interfaces, and other resources.	
Types and features of middleware		(1) The basic architecture, role, and features of middleware. (2) Web servers. (3) Application servers. (4) Databases. (5) The current status of implementation of OSS middleware. (6) Introduce typical OSS compatibles with such middleware.	
Computer system configuration			
Examples of system architecture utilization			
Web system architecture			
Infrastructure design casework using OSS			

Open source system architecture creation			
Hardware as an OSS operating environment			
Future trends in open source architecture			

SKILL CATEGORY NAME	Basic		SKILL CATEGORY NO.	1
SKILL NAME	Distributed Architecture		SKILL NO.	4
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Distributed architecture	II			
Name management	II			
Duplicate management	II			
Fault tolerance	II			
Secure channels	II			
CORBA	II			
Web services	II			
Distributed transactions	II			
Peer-to-peer architecture	II			
Mobility	II			
Consistency	II			
Large-scale distributed systems	II			

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	Concept of Linux and Basic Operations		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS		CODE NO.
The outline of Linux	I	(1) UNIX and Linux, and the history of Linux development. (2) The role of operating system (OS) and the basic idea of UNIX. (3) The relations between UNIX and Linux. (4) The history of Linux development.		
	II	(1) Linux development model and points to remember in using the model. (2) The development style using the bazaar model–a feature of the Linux development model. Concerning the use of Linux and applications running on Linux, present cases related to licenses and business. Explain the general use of Linux applications. (3) The substance, definitions, and features of Linux distribution. (4) Examples of typical distribution systems. (5) The features and types of each distribution system.		
System management	I	(1) The concept of the user. (2) The log-in and log-out procedure. (3) The basic operation of UNIX/Linux using the B shell (bash). (4) The basic idea of dialogs with OS using a shell. (5) Online manuals. (6) Man command. (7) Info command. (8) Practical operations such as top, ps etc. (9) Editor (vi emacs). (10) Pager (more less). (11) Input-output stream. (12) The idea of redirect and pipe. (1) Tree structure of the file system and its operation.		

		(2) The file system as a tree structure. (3) Many types of file systems. (4) The concept and features of the file system. (5) Typical commands and their usage in basic file/ directory operation.	
Multitasking		(1) The idea of multitasking as a basis for processing on UNIX/Linux. (2) The difference between single-tasking and multitasking. (3) The difference between single-users and multi-users. (4) Time-sharing systems and multitasking.	
File systems	I	(1) Multiple-users and groups. (2) Permission by the users and groups.	
	II	(1) Access lists (POSIX ACL).	
Data saving and backup			
Shell scripts and development environment			
The basics of networks			

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	Linux System Management		SKILL NO.	3
TOPICS	LEVEL	SUBTOPICS		CODE NO.
The outline of Linux system management work	I	(1) Role and management operations of the system administrator. (2) Outline of system management operations. (3) The purpose, necessity, and types of system management operations. (4) The roles and authority of the system administrator. (5) The mindset required of the administrator.		
Linux system and server management	I	(1) Implementation of a Linux system. (2) Linux implementation tasks. (3) An actual implementation procedure. (4) The setup procedure. (5) The implementation and management procedure for application packages. (6) Precautions to be taken in implementation. (7) Startup and shutdown of the Linux system. (8) The startup and shutdown procedure for the Linux system. (9) Init process. (10) Runlevel. (11) Basic knowledge of the startup and shutdown of services.		
Linux system and file/disk management				
Linux system and user management				

Linux system management, and backup and log operation management			
Linux system and resource management			
Linux system and kernel management			
Linux system and network management			
Linux system and routing management			
Linux system management, and DHCP construction and operation			
Linux system management, and FTP construction and operation			
Linux system management, and NFS construction and operation			
Linux system management, and Samba building and operation			
Linux system management and basic operation work troubleshooting	II		
	III		
Linux system management and network troubleshooting	II		
	III		

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	Linux System Programming		SKILL NO.	4
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Linux system management and network troubleshooting	II			
Shell programming	II			
File input/output programming	I			
File systems	I			
UNIX environments	I			
The use of libraries and their construction procedures	I			
Data management	II			
Software development environments	II			
Debugging	I			
Processes and threads	II			
Signals	II			
Communications between processes and pipes	II			
Input/output to/from terminal equipment	II			
Semaphores, shared memory and message queues	II			
Network programming	II			

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	Network Server Management		SKILL NO.	5
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Functions and features of network servers	I	(1) Role of the server and communication protocols. (2) The role and basic mechanism of the servers. (3) The functions and features of network servers. (4) Communication protocols. (5) Basic concepts of client-server systems. (6) Meaning of the network address, domain name, and host. (7) The structure of the network address (IP address). (8) Domain name, and host name. (9) Network byte order. (10) Uniform Resource Identifier (URI).		
Server system installation	I	(1) An orderly outline and work procedure for the introduction of general network servers. (2) The acquisition of OS and required software packages, implementation, configuration of the startup environment, network configurations, service content settings, and the startup and shutdown of the server system.		
Name server installation	I	(1) Give an outline of Domain Name System (DNS) and the mechanism of the DNS server and the DNS protocol to offer DNS services. Also provide related information concerning such topics as the history of, and background to, DNS server implementation based on OSS. (2) The procedure for introducing, configuring, and setting up a Linux-based name server. (3) The configuration file.		
Web server installation	I	(1) Mechanism of the Web server. (2) The functions and roles of the Web server. (3) The execution and extension of applications with CGI. (4) The history of, and background to, Web server implementation with OSS.		

		(5) The outline of Hyper-Text Transport Protocol (HTTP) and communication methods. (6) Procedure for introducing, configuring, and setting up a Linux-based Web server. (7) The configuration file.	
Super server installation	II	(1) inetd (2) xinetd (3) tcpwrapper	
Proxy server installation	II	(1) Squid (2) Apache proxy	
	III	(1) Reverse proxy. (2) Proxy authentication. (3) NTLM authentication.	
Details of other network server installation work and its work procedures	II		
Routing processing and filtering processing implementation by network servers	II		
Internet connection via network servers	II		
Server operation management tasks	II		
Log management details and procedures	II		
Linux server security	II		
Linux service security	II		
Functions and implementation of secure	II		

OSes			

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	Cluster System Architecture		SKILL NO.	6
TOPICS	LEVEL	SUBTOPICS		CODE NO.
A general introduction to cluster systems	I	(1) The outline and features of the High Availability (HA) cluster and the High Performance Computing (HPC) cluster. (2) Compare the two cluster systems in terms of the purpose and position. (3) The load distribution technology as a key HA cluster technology, explain its concepts, types, configuration, and components. Introduce the practical application of the load balancer, which enables load distribution. Describe the configuration and features of each type of load balancer. (4) Introduce typical algorithms, such as a round robin algorithm, that enable load distribution. Explain a practical procedure for the algorithm. Address the technique to monitor servers for their state of life or death. Explain tasks to be performed during the course of operation of the HA cluster.		
HA clusters	I	(1) Realization of a load balancer by the Linux Virtual Server (LVS). (2) The components of the system. Provide a practical procedure for building an IPVS kernel, ipvsadm and keepalived, a testing environment, and a Direct Server Return (DSR) system. (3) Redundancy by the Virtual Router Redundancy Protocol (VRRP). (4) The Virtual Router Redundancy Protocol (VRRP), which is used to provide redundancy for routers. Explain the ways to configure a network and make keepalived settings achieve redundancy. Also give practical advice on keepalived management.		
Computer simulation		(1) Supercomputer: history and realization by clusters. (2) The outline, history, and recent trend of the supercomputer as part of the scientific and technological infrastructure. Explain how an increasing number of supercomputers have been developed in recent years by the use of clusters. Introduce some PC-cluster projects.		

A general introduction to parallel programming	I	(1) Types and configuration of parallel computers. (2) The principles of parallel programming. (3) Types of parallel computers and differences between them in configuration for processor, memory binding method, and bus topology. (4) Parallel processing and the basics of programming for parallelization. (5) Typical algorithms to demonstrate parallelization techniques. (6) The parallel programming environment and tools available for parallel programming.	
Beowulf PC cluster construction	II		
SCore clusters	II		
PC cluster-related technology	II		
Grid computing	II		

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	Concurrent System Programming		SKILL NO.	7
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Introduction to concurrent systems (incl. HA, HPC cluster)	I	(1) Introduction to concurrent system: A. SMP system B. HA, HPC cluster system C. Grid computing		
Parallel processes	II	(1) Parallel process. (2) Concurrency. (3) Synchronization. (4) Mutual Exclusion. (5) Deadlock. (6) Generic synchronization primitives: A. Interrupt control B. Mutex primitives (lock, semaphore) C. Monitor		
Linux IPC mechanisms	II	(1) Using Linux IPC mechanisms: A. Message queue B. Shared memory C. Pipe (2) UNIX socket		
Lightweight processes – POSIX threads	II	(1) pthread programming: A. Create / exit B. Join (2) Shared data handling with pthread: A. Mutex B. Semaphore C. Conditional wait		

Parallel programming	III	(1) Parallel programming with standard interface: A. openMP B. MPI (2) Monitoring and debugging of parallel program.	
Asynchronous events	II	(1) Signal handling. (2) setjmp/longjmp	

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	JavaEE Application Server		SKILL NO.	8
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Understanding Java EE	I	- Java EE Architecture overview - Guide to OSS JavaEE Application Server: ■ Apache HTTP Server, Tomcat, JBoss, Gernimo, JOnAS and so on ■ OSS Framework: Struts, Spring, Hibernate, iBatis and so on - Installing and configuring Apache, Tomcat, JBoss		2-8-1-I
Web programming with JSP/Servlet	II	- Developing a basic Java Servlet - Developing a View Component - Developing a Controller Component - Developing Dynamic Forms - Sharing Application Resources Using the Servlet Context - Designing the Business Tier - Developing Web Applications Using Struts - Developing Web Applications Using Session Management - Using Filters in Web Applications - Integrating Web Applications With Databases: ■ JDBC overview ■ JDBC Statement/Prepared Statement ■ JDBC Transaction ■ JDBC Exception ■ Connection Pool ■ Introduction to JDBC 2.0 and 3.0 - Developing JSP Pages		2-8-2-II

		- Developing JSP Pages Using Custom Tags - Developing Web Applications Using Struts Action Forms - Building Reusable Web Presentation Components	
EJB programming	II	- Overview of developing Java EE component - Developing EJB 3.0 component - Developing Entity class using Java Persistence API(JPA) - Developing Model2-style web component - Developing Transactional component - Developing Java Messaging Service(JMS) component - Developing web services component using EJB 3.0 component - Overview of Security API	2-8-3-II
JSF programming	III	- Introduction to JSF - Standard Features: ■ Creating backing beans and using managed beans ■ Exploring the standard components ■ Internationalization, validators, and converters ■ Using Facelets - Application Development: ■ Developing with JSF technology ■ Inside the JSF Architecture ■ Writing custom components, validators, and converters - Extensions and Integration: ■ Component frameworks overview ■ Using Apache Shale (appendix) ■ Migrating from Struts (appendix) ■ Using JBoss Seam (appendix)	2-8-4-III

		■ Summary and future directions	
Programming with OSS JavaEE Framework	III	- Understanding Object-oriented Design Patterns - Understanding Software architecture - Understanding JavaEE architecture layer - OSS JavaEE Framework overview - Classification of OSS JavaEE Framework - Supporting Framework(Maven, Log4j, JUnit) overview - Spring Framework overview - DI, SpringMVC, SpringDAO, AOP and so on - ORM Framework overview - ORM Framework (Hibernate, iBatis) overview	2-8-5-III
J2EE on JBoss	III	- JBoss: ■ Overview of JBoss Application Server ■ Highlights of JBoss AS ■ JBoss AS Architecture ■ JBoss AS Requirements - Installation - Directory Structure - Starting/Stopping - Deployment: ■ J2EE development-deployment life-cycle ■ J2EE deployment descriptors ■ Deployment on JBoss AS ■ Deployment Dependencies ■ Hot vs. Cold Deployment ■ Looking at conf/jboss-service.xml and XMBeans	2-8-6-III

		- JBoss-IDE:
		  ■ Overview of JBoss IDE plug-ins for Eclipse
		  ■ Configuring JBoss IDE
		  ■ Controlling JBoss AS from Eclipse
		- Web Tier:
		  ■ Java Servlets
		  ■ Welcome files
		  ◆ Error documents
		  ■ Introduction to Servlet Filters
		  ◆ Life-cycle
		  ◆ API
		  ◆ Defining and mapping in WEB-INF/web.xml
		  ■ Java Server Pages
		  ■ Tomcat Web Container
		  ◆ Overview of Tomcat
		  ◆ Architecture
		  ◆ Configuration (jboss-service.xml and server.xml)
		  ◆ Tomcat's web.xml
		  ◆ Serving static content
		  ◆ Virtual hosting
		  ◆ Web access logging
		- JNDI:
		  ◆ Overview of JNDI
		  ◆ Role of JNDI in J2EE
		  ◆ JNDI API
		  ◆ Using JNDI
		  ◆ JNDI resources
		  ◆ JNDI on JBoss AS
		- JavaMail:	

		■ Overview of JavaMail ■ JavaMail API ■ JavaMail service on JBoss AS ■ Sending (SMTP) and fetching (IMAP/POP) mail - JMX: ■ Overview of Java Management Extensions ■ JMX Architecture (layers) ■ JMX MBeans ■ JMX on JBoss AS ■ JMX Console ■ Twiddle tool (JMX command-line client) ■ Developing MBeans through JBoss IDE and XDoclet ■ Packaging service archive (SAR) files ■ Web Console ■ JBoss Monitoring - Class Loading: ■ Java Class Identity ■ J2EE Class Loading Requirements ■ Class Loading on JBoss ■ Default Class Search Order ■ Scoped Class Search Order ■ Log4j Issues ■ Common Problems With Class Loading - Database Integration: ■ Managed Database Connection Pools ■ Referencing database connection pool resources ■ Installing JDBC drivers ■ Defining database resources in JBoss AS ■ Resource Mapping	

		■ MySQL example ■ JBoss embedded Hypersonic database ■ Detecting DB Connection Leaks - JMS: ■ Overview of Java Messaging Service ■ JMS Architecture ■ JMS Messaging Domains (Point-to-Point, Publish and Subscribe) ■ JMS Message Consumption ■ JMS API ■ Developing with JMS ■ JMS on JBoss AS ◆ Configuration ◆ Creating destinations (queue and topic) - EJB - Transactions: ■ Overview of Transactions (ACID properties) ■ Resource Locking (pessimistic vs. optimistic) ■ Distributed Transaction Components (JTA API) ■ Two-phase XA protocol ■ Heuristic Exceptions ■ Transactions on JBoss AS ■ Container-Managed Transactions (CMT) ■ User Transactions - Web Services: ■ Web Services on JBoss AS ■ Servlet-based web service example ◆ Creating end-point ◆ Generating descriptors (including WSDL) using WSDP ◆ Packaging and deploying	

		■ Creating web service client - Security: ■ Filtering Clients by Source (IP address or hostname) ■ Authentication and Authorization using JAAS ◆ Role-based declarative security ◆ Requiring authentication and authorization in deployment descriptors ◆ JBoss plain-text login module ◆ JBoss database login module (example with MySQL) ◆ Linking to security domain ◆ Securing passwords with MD5 ◆ FORM-based login (including the handling of errors) ◆ Configuring JBoss AS for SSL ◆ Generating SSL certificates ◆ Tomcat's SSL connector ◆ Requiring SSL in web applications ■ Securing JBoss AS ◆ Running with low-privileges ◆ File-system security ◆ Securing console applications/tools and services ◆ Running with Java Security Manager ◆ Running behind a firewall	

SKILL CATEGORY NAME	Network		SKILL CATEGORY NO.	3
SKILL NAME	Network Architecture		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS		CODE NO.
The concept and mechanism of open networks				
Communications forms and protocols				
The mechanism of Internet communications				
The mechanism of LAN networks				
Types of wireless networks and the mechanism of communications				
Open network communications specifications				
The mechanism of IP networks				
The mechanism of routing				
Routing protocol specifications				
The mechanism of TCP				
Communications protocol operation check				
The mechanism of TCP applications	II	(1) Web communications includes http, https. (2) FTP, telnet. (3) ssh		

New network architecture	III		

SKILL CATEGORY NAME	Network		SKILL CATEGORY NO.	3
SKILL NAME	Network Management		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS		CODE NO.
The outline of network system operation				
Individual items and details of network management				
Individual items and details of network capacity management				
Individual items and details of network performance management				
TCP/IP management				
Network server operation management practice				
Network server operation management practice				
Network hardware operation management				
The outline of network management protocols		(1) snmp (2) mib		
Exercising network management using MRTGs				
Network operation design				
Operation management practical procedures and its system				
WAN operation management				
Network fault management				

Network troubleshooting			

SKILL CATEGORY NAME	Network		SKILL CATEGORY NO.	3
SKILL NAME	Network Programming		SKILL NO.	3
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Data Communication Basic	I	- Fundamentals of data communication technologies: ■ Dedicated / switched / virtual circuits ■ Analog and digital transmission ■ Bandwidth and latency ■ Transmission media - Connectivity and inter-networking: ■ Network topologies ■ LAN equipment ■ Ethernet / Wireless network - Packet / protocol / interface		3-3-1-I
Network architecture and Internet	I	- Layered network architecture: ■ OSI 7 layer reference model - Internet architecture: ■ Definition and topologies ■ Addressing - Domain name system - Internet protocols: ■ TCP/IP protocol stacks ■ Application level protocols		3-3-2-I

Linux network system architecture	I	- Linux network system architecture: ■ OSI Layer mapping to Linux network system - Linux network commands: ■ netstat / route / ifconfig / ping - Network configuration files under /etc: ■ rc scripts ■ Super server (xinetd) configuration - Linux network security basics	3-3-3-I
TCP/IP protocol internals	II	- TCP/IP protocols: ■ ARP / IP / ICMP / TCP / UDP ■ TCP flow control / error recovery mechanism - Application layer protocols: ■ telnet / ftp / http / mail	3-3-4-II
Client server architecture	I	- Client - server architecture: ■ Basic operation scenarios ■ Naming / connection / transfer / disconnection - Stateful / stateless server - Peer-to-peer architecture	3-3-5-I

TCP Socket programming	II	- Using bsd socket interface: ■ Creating and destroy ■ Byte ordering - TCP client-server programming: ■ connect() ■ bind() ■ listen() ■ accept() ■ recv() / send() - Name resolving: ■ Hostname and IP address	3-3-6-II
	III	- Socket options	3-3-6-III
UDP Socket programming	II	- UDP message transfer: ■ recvfrom() / sendto() - UDP client-server programming: ■ bind() connect()	3-3-7-II
	III	- UDP multicasting	3-3-7-III
Server programming	II	- Supporting multiple servers with fork() - Server using super-server (xinetd)	3-3-8-II
Synchronous (blocking), asynchronous (non-blocking) interface	II	- Blocking I/O, non-blocking I/O - Non-blocking socket programming: ■ select() / pselect() ■ poll() / ppoll() - Multiple socket management - Multithread socket application	3-3-9-II

Linux IPC mechanisms	II	- Using Linux IPC mechanisms: ■ Message queue ■ Shared memory ■ Pipe ■ UNIX socket	3-3-10-II
Network programming with security	III	- Programming with OpenSSL	3-3-11-III
Interfacing to OSS P2P, VOIP	III	- P2P programming: ■ Programming with gtk-gnutella - VOIP programming - Programming with Openh323	3-3-12-III

SKILL CATEGORY NAME	Programming		SKILL CATEGORY NO.	4
SKILL NAME	Java		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS		CODE NO.
The basics of Java				
Fundamental structure of Java language				
Advantages of object-oriented programming				
Procedures for application development in Java				
Network programming in Java				
The outline of Web application development according to Servlet/JSP/JDBC	I			
Database access according to JDBC	II			
MVC models	I			
Application development according to EJB	II			
Object-oriented system analysis/design/implementation practice technology	II			
Development procedures according to design patterns	II			
Java performance tuning	II			

SKILL CATEGORY NAME	Programming		SKILL CATEGORY NO.	4
SKILL NAME	C		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS		CODE NO.
The basics of C	I	(1) Compile and gcc command line options. (2) Optimizations. (3) Directives. (4) Variables and variable types. (5) Operators.		
Fundamental structure of C	I	(1) Loop. (2) Branch with condition. (3) Function call. (4) Header files and includes. (5) Prototype definitions. (6) Standard libraries.		
	II	(1) Berkley db libraries. (2) Compression libraries.		
Character string operation				
Functions				
Pointers				
Structures				
Console input/output				
File management				
Data structure				

SKILL CATEGORY NAME	Programming		SKILL CATEGORY NO.	4
SKILL NAME	C++		SKILL NO.	3
TOPICS	LEVEL	SUBTOPICS		CODE NO.
C++ Basic Programming	I	(1) Class & Object. (2) Functions and Member functions. (3) Inheritance & Containment. (4) Object Pointer, Array & Structure. (5) Dynamic allocation (new, delete). (6) Polymorphism (Virtual Function, Overriding). (7) Exception Handling. (8) Namespace. (9) Template.		4-3-1-I
STL	II	(1) Container, Sequence, Associative Container & allocator. (2) Refinement (reflexivity, containment, transitivity). (3) Trait. (4) Regular type (assignable, default constructible, equality comparable, less than comparable). (5) Iterator (trivial iterator, input iterator, output iterator, forward iterator, bidirectional iterator, random access iterator). (6) Function Object (generator, adaptable generator, predicate).		4-3-2-II

SKILL CATEGORY NAME	Programming		SKILL CATEGORY NO.	4
SKILL NAME	Light Weight Languages		SKILL NO.	4
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Use of development libraries				
Fundamental structure of Perl				
Fundamental structure of PHP				
Fundamental structure of Python				
Fundamental structure of Ruby				
Development process procedures using development frameworks				
Object-oriented programming				
Embedded classes				
concept of Ruby on Rails	I			
Database application development				
Plug-in installation and development				
Open source system customization				

SKILL CATEGORY NAME	Programming		SKILL CATEGORY NO.	4
SKILL NAME	GUI		SKILL NO.	5
TOPICS	LEVEL	SUBTOPICS		CODE NO.
GTK	II	- GTK Programming Scheme - Initializing GTK - GtkWidget & GtkWindow & Main Loop - Widget Hierarchy - Widget Properties - Signal & Callbacks, Events - Container Widgets (GtkContainer, Boxes, Panes, Tables, Expanders, Handle Boxes, Notebooks, Event Boxes) - Buttons - Dialogs (Built-in Dialogs, Dialogs with Multiple Pages) - Text View Widget (Scrolled Windows, Text Views, Text Iterators and Marks, Text Tags, Inserting Image & Child Widgets, GtkSourceView) - Tree View Widget (GtkListStore, GtkTreeStore, Referencing Rows, Adding Rows and Handling Selections, Editable Text Renderers, Cell Data Functions, Cell Renderers) - Menus and Toolbars (Pop-up Menus, Keyboard Accelerators, Status Bar Hints, Menu Bars, Toolbars, Toolbar Items, Dynamic Menu Creation, Custom Stock Items). Dynamic User Interfaces (User Interface Design, The Glade User Interface Builder, Using Libglade) - Using gtk_config		4-5-1-II

QT	II	- Qt programming Scheme (signal, slot, emit, Q_OBJECT) - Qt Class Hierarchy - Main Windows (Qmain Window, Menus and Toolbars, Status Bar, Multiple Documents, Splasy Screens) - Base Classes (Qobject, Qstring and Other Classes, Inheritance Hierarchy) - Layouts (Manual Layout, Automatic Layout, Splitter, Stacked Layouts) - Dialogs (Modal Dialogs, Non-modal Dialogs, Semi-modal Dialogs, Avoiding Bloated Dialogs, Ready-made Dialogs in Qt) - Qtable Widget & Qtable WidgetItem - Item View Classes - Container Classes - Custom Widgets (Qwidget, Qt Designed, Double Buffering) - Events and the Clipboard (Event Loop and Handler, Handling Events, Event Filters, Drag and Drop) - 2D and 3D Graphics - Drag and Drop - QtSql Module - Input/Output Interfaces - Threading with Qthread - Handling XML with QtXml - Using qmake - Qt Designer (layout and form design tool)	4-5-2-II

SKILL CATEGORY NAME	Programming		SKILL CATEGORY NO.	4
SKILL NAME	Web Programming		SKILL NO.	6
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Understanding Web architecture and web 2.0 technology	I	- Web architecture overview - Introduction to web 2.0: ■ Web 2.0 trends overview ■ Social web ■ Folksnomy & Tagging ■ Mashup ■ RSS and Atom ■ Google map API (Geospatial Web) ■ Semantic Web ■ Blog		4-6-1-I
Web programming with PHP	II	- Introduction to PHP - Variables and Expressions in PHP - PHP Operators - Conditional Tests and Events in PHP - PHP Flow Control - Functions in PHP - Arrays - Object-Oriented Programming in PHP - Adding and Accessing Dynamic Content - Cookies - Sessions - File and Directory Access Using PHP		4-6-2-II

		- String Manipulation and Regular Expressions - Managing Date and Time - PHP Debugging - MySQL with PHP	
Web programming with JSP/Servlet	II	- Developing a basic Java Servlet - Developing a View Component - Developing a Controller Component - Developing Dynamic Forms - Sharing Application Resources Using the Servlet Context - Designing the Business Tier - Developing Web Applications Using Struts - Developing Web Applications Using Session Management - Using Filters in Web Applications - Integrating Web Applications With Databases - Developing JSP Pages - Developing JSP Pages Using Custom Tags - Developing Web Applications Using Struts Action Forms - Building Reusable Web Presentation Components	4-6-3-II
Web programming with Ruby	I	- Ruby Overview: ■ Objects ■ Classes ■ Core Ruby ■ Ruby Standard Library ■ Control Structures	4-6-4-I

		■ Scope ■ Blocks ■ Modules - Ruby for Rails: ■ Rake ■ Test/unit - Demystifying Rails: ■ The Console ■ Models, Controllers ■ Associations ■ Migrations ■ Views ■ Core Ruby extensions - Test-Driving your Rails App: ■ Beyond Scaffolding ■ Unit and Functional Tests ■ Integration Tests - Advanced Controllers and Models: ■ Validations ■ User Authentication ■ TDD'ing Actions ■ Evolving Associations - Rails Power Tools: ■ Mock Objects ■ Named Routes ■ Advanced Integration Testing ■ Rails Plugins - Ajax on Rails:	

		■ Helpers and RJS ■ Ajax with Prototype and Scriptaculous ■ Ajaxifying your Rails App ■ Advanced Ajax on Rails	
XML programming with Java	II	- Overview XML - XML Namespace - DTD - Schema - XPath Syntax - XSL - XML, SAX, DOM and JAXP - Java Programming with namespace-aware parser using JAXP - Java Programming using SAX API - Java Programming using DOA API	4-6-7-II
Developing Java Web Services	II	- Web Services overview - Introducing Java technology & platform about Web Services - SOAP - SAAJ - WSDL - Service Registry - JAX-RPC - Overview of Web Services Security - Design guideline of Web Services Security	4-6-8-II
Web Application	II	- Introduction to Ajax:	4-6-9-II

Development with Ajax		■ What is Ajax (where to use it, and why does it matter)? ■ Synchronous and asynchronous interaction ■ The XmlHTTPRequest object ■ Retrieving data as text and as XML ■ Using HTTP methods, headers and parameters ■ Asynchronous callback handlers - Ajax Design Basics: ■ Retrieving content ■ Retrieving executable scripts ■ Retrieving data: text, XML, JSON ■ Refactoring the XHR object - Prototype: ■ Extending the core language ■ New coding idioms for JavaScript technology ■ Ajax Helper classes ■ HTML Form and DOM helpers	
Web 2.0 programming with Java	III	- Component Models: JSF, Tapestry, Atlas - Exposing Server Objects: DWR, SAJAX - Security concerns: ■ Restricting access ■ Protecting data ■ Web 2.0 architecture: Services Integration and Mashups - Consuming third-party services: ■ RSS ■ Google maps ■ Web services - Exposing a third-party API: ■ SOAP	4-6-10-III

		■ XML-RPC ■ REST - Code base management: ■ Profiling JavaScript Code ■ Testing Ajax applications	

SKILL CATEGORY NAME	Multimedia System		SKILL CATEGORY NO.	5
SKILL NAME	Multimedia Service Platform		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS		CODE NO.
The basics of Multimedia Service	I	- An introduction to Multimedia Service - Multimedia Service Platform Overview - Streaming Server (Broadcasting & VOD/AOD System) - Upload/Download Server - Client software		5-1-1-I
	II	- Codecs: ・Snow ・FFV1 ・ATRAC3 ・H.261, H.263 and h.264/MPEG-4 AVC ・Indeo 2 and 3 ・QDesign Music Codec 2, used by many QuickTime movies prior to QuickTime 7 ・Smacker video ・Sorenson 3 Codec used by many QuickTime movies ・Theora (together with Vorbis makes a base for the .ogg format) ・Truespeech ・TXD ・VP5 and VP6 ・Vorbis ・Windows Media Audio ・Some Windows Media Video codecs, including WMV1, WMV2 and WMV3 - Multimedia File Formats: ・ASF, AVI, BFI, IFF, RL2, FLV, Matroska, Maxis XA, TXD ・MSN Webcam stream, MPEG transport stream		5-1-1-II

Implementation of multimedia service platform using FFmpeg	II	- Introduction to FFMPEG - ffmpeg: command line tool to convert one video file format to another - ffserver: HTTP multimedia streaming server for live broadcasts - ffplay: simple media player based on SDL and on the FFmpeg libraries - libavcodec: all the FFmpeg audio/video encoders and decoders - libavformat: demuxers and muxers for audio/video container formats - libavutil: helper library containing routines common to different parts of FFmpeg - libpostproc: video postprocessing routines	5-1-2-II
	III	- Multimedia Service Architecture based FFmpeg - Video and Audio grabbing - X11 grabbing - Video and Audio file format conversion - Uploading & Content Managements - Broadcasting Service - VOD Service	5-1-2-III

SKILL CATEGORY NAME	Multimedia System		SKILL CATEGORY NO.	5
SKILL NAME	Multimedia Programming		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Understanding Multimedia Programming	I	- Multimedia Programming Basic - Multimedia Application Implementation on Web (API Programming Skill) - Display Objects - Text & Image - Geometry Objects (Point, Line, Rectangle, Polygon etc) - Introduce Programming Tools (Video4Linux, OpenSoundSystem etc)		5-2-1-I
	II	- Basics of codecs - Event Processing - Style & Theme - Effects - Data handling & Using XML - Video & Camera handling - Using Audio & Microphone - Optical media (CD, DVD) interface - Print Skill - JPEG library - FFMPEG library		5-2-1-II

Video4Linux	II	- An introduction to Video4Linux - Graphics cards with TV Tuner and/or Capture facilities - TV cards and drivers - Supported tuners - Video via PCI - Video via PCI Express - Video via USB - Firewire - Loopback - Radio - Remote controllers - Scanners	5-2-2-II
	III	- An introduction to API - Registration and open() - Basic ioctl() handling - Inputs and Outputs - Colors and formats - Format Negotiation - Basic Frame I/O - Streaming I/O - Controls	5-2-2-III

OpenSoundSystem	II	- An introduction to OSS (Open Sound System) - OSS in relation to ALSA (Advanced Linux Sound Architecture) - OSS/3D - Digital voice device (Linear Encoding) - Mixer - Synthesizer - Music (mu-law Encoding)	5-2-3-II
	III	- Audio Programming - Opening the Sound Device - Sampling Parameters - Audio Format - Channels - Sampling Rate - Half & Full Duplex - Input/Output - Multitasking	5-2-3-III

SKILL CATEGORY NAME	Development System		SKILL CATEGORY NO.	6
SKILL NAME	Development Frameworks		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Introduction of development framework				
categories and features of development frameworks				
OSS frameworks for web applications				
The outline of free Web containers/J2EE containers				
Open source development tools				
Development process on deployment of frameworks				

SKILL CATEGORY NAME	Development System		SKILL CATEGORY NO.	6
SKILL NAME	Development Tools		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Development flow and tools				
The outline of software development environments				
The outline of software application development in the Linux development environments				
Version management tool utilization				
Program debugging environments using debuggers				
Debugging using kernel debuggers				
Debugging using bug tracking systems				
Types and functions of open source development tools				
Development procedures in integrated development environments				
Types and features of open source integrated development environments				
Workshops on software				

development in Linux development environments			
The outline of software development support tools in Linux development environments			
Software development tool evaluation			
Software development using Eclipse			

SKILL CATEGORY NAME	Development System		SKILL CATEGORY NO.	6
SKILL NAME	Integrated Development Environment (IDE)		SKILL NO.	3
TOPICS	LEVEL	SUBTOPICS		CODE NO.
<TBD>				

SKILL CATEGORY NAME	Security		SKILL CATEGORY NO.	7
SKILL NAME	Encryption		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Security functions and encryption positioning				
Encryption systems/common key cipher systems				
Encryption systems/public key cipher systems				
Encryption application systems in information systems				
The mechanism of digital certificates				
OSS utilization scenes and encryption				
Wireless LAN encryption				
Authentication and encryption				
Encrypted communications using IPsec				
Tunneling using SSH				
The mechanism of SSL protocols				
VPN communications establishment				
The mechanism of PKI (public key encryption infrastructure)				
Authentication infrastructure building hands-on training				
Encryption – future utilization scenes and issues				

SKILL CATEGORY NAME	Security		SKILL CATEGORY NO.	7
SKILL NAME	Network Security		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS		CODE NO.
<TBD>				

SKILL CATEGORY NAME	Security		SKILL CATEGORY NO.	7
SKILL NAME	OS Security		SKILL NO.	3
TOPICS	LEVEL	SUBTOPICS		CODE NO.
<TBD>				

SKILL CATEGORY NAME	RDB		SKILL CATEGORY NO.	8
SKILL NAME	Basic Skills in RDB		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS		CODE NO.
<TBD>				

SKILL CATEGORY NAME	RDB		SKILL CATEGORY NO.	8
SKILL NAME	RDB System Management		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS		CODE NO.
<TDB>				

SKILL CATEGORY NAME	RDB		SKILL CATEGORY NO.	8
SKILL NAME	RDB Applications Development		SKILL NO.	3
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Introduction to RDB applications development	I	• RDB application overview • RDB application development methodology		8-3-1-I
DB system architecture	I	• Client/Server database system architecture • Web database system architecture		8-3-2-I
RDB design and creation	I	• RDB design overview • RDB design methodology		8-3-3-I
	II	• ER modeling • Relational DB design and normalization • Workshop on database design • Database creation		8-3-3-II
Writing queries with SQL and optimizing it	I	• SQL overview and syntax • Writing various queries with SQL		8-3-4-I
	II	• Optimization of SQL statements		8-3-4-II
Database programming with C/C++	II	• Introduction to DB programming with C/C++ • MySQL C/C++ API overview • Connecting to database • Querying database • Retrieving the query results • Inserting, deleting, and updating data in database • Using prepared statement • Handling of Multiple Statement Execution • Using transaction • Disconnecting from database		8-3-5-II
	III	• Making a threaded client		8-3-5-III
DB programming with JDBC and JSP/Servlet	II	- JDBC and/or JSP/Servlet programming overview - JDBC driver types		8-3-6-II

		- Installing and configuring JDBC and TOMCAT environment • JDBC interface overview • Setting up a connection • Querying database • Retrieving the query results • Inserting, deleting, and updating data in database • Using transaction • Using Cookie and session • Disconnecting from database • JDBC exception handling • Using prepared statements and stored procedure • O/RM (Object-relational Mapping) • Samples of DB application written using JDBC and JSP/Servlet • Practices in developing RDB applications using JDBC and JSP/Servlet	
DB programming with PHP	II	• Overview of PHP programming • Installing and configuring PHP and Apache • Setting up a connection • Querying database • Retrieving the query results • Inserting, deleting, and updating data in database • Using transaction • Disconnecting from database • Using prepared statements and stored procedures • Samples of Web DB application written by PHP • Practices in developing RDB applications using PHP	8-3-7-II
Development of XML DB applications	II	• Introduction to XML DB applications overview • XML and XML related tools	8-3-8-II
	III	• Mapping between tables in RDB and elements in XML • Relationships in RDB and containment & pointers in XML • Modeling exercise with example	8-3-8-

		• Migrating an existing RDB to XML • Migrating existing XML documents to RDB • Querying RDB and writing XML by adding element tags yourself • Querying RDB and writing XML by using a utility module • Extracting records from XML with XML-parser and storing them into RDB • Extracting records from XML with XML-Xpath and storing them into RDB • Samples of XML DB applications • Practices in developing XML DB applications	