
Knowledge and skills required for OKTET Labs engineers

Piotr Lavrov <Piotr.Lavrov1@oktetlabs.ru>

Table of Contents

1. 1. Linux usage	1
2. 2. Development tools	1
3. 3. C language	2
4. 4. Network concepts	3
5. 5. Operating Systems concepts	3
6. 6. Documentation	4
7. 7. Internet technology	4
8. 8. Programming concepts	4
9. 9. Communication and language skills	4

1. 1. Linux usage

You need to have user-level access to Linux system (in some situations, OKTET Labs may provide access to it's host). It is highly recommended to install Linux on the home PC by itself (you may ask guru to assist you).

Know how to use bash facilities: loops, if, test, pipes, environment variables, redirections.

Background and foreground jobs; daemons.

Understand Linux FS from user point of view: unix file permissions model, file ownership, ordinary files, devices, hard and symbolic links.

Regular expression language.

cat, echo, tee, less, strings, chmod, chown, find, grep, diff, patch, tar, gzip, bzip2, xargs, sort, uniq.

Ability to use one of "plain text" editors (vim, emacs, joe, mcedit, or other).

Understanding of "package" concept in Linux distribution. Knowledge how to deal with packages (installation, uninstallation, query) in your favourite Linux distribution.

ssh, scp, ftp, telnet.

2. 2. Development tools

cvs or svn

Using gcc as pre-processor, compiler and linker. Options for enabling/disabling warnings, debugging information, optimisations.

~~binutils: ar, nm, objdump, objcopy.~~

`gdb`: run program, set break points, print variables values, set arguments, backtrace, up, down.

`valgrind`, `strace`, `ptrace`.

`make`: targets, rules, variables, implicit rules, traditional names for targets and variables.

Know how to build typical UNIX software package from source code tree. `configure` script, it's standard options. Source, build and installation trees. You should try to build and install several packages.

`autoconf` & `automake`: know how to write simple `configure.ac` and `Makefile.am`. Know what is `aclocal` and `autoheader`.

GNU C. `gcc` as a compiler driver. Preprocessor, compiler, assembler, linker.

CPU architecture simulators - `gdb` integrated and standalone. Knowledge of concepts. Using simulators with `gdb`. You should try to build and run sample program using simulator.

3. 3. C language

Data types. `struct` and `union`. Pointers and arrays.

Variables. Local, static and global variables. Variable initialization.

Type casting. Explicit and implicit casting.

Functions. Passing arguments to the functions. Passing by value vs. passing by reference.

Statements. `for`, `while`, `if`, `do while`, `switch`, `goto`, `break`, `continue`.

Arithmetical expression. Operators: arithmetical, logical, bitwise, assignment, increment, decrement, ternary conditional. Operator precedence (Understanding what does it mean, and ideas how/why they are arranged. No need to know operator precedence table exactly). Undefined behaviour.

Preprocessor. Preprocessor directives. `.h` and `.c` files.

Scope of variable and functions. Storage class specifiers: `const`, `volatile`.

Literals: integer, character, strings, unsigned int, long, floating points.

String representation. String manipulations.

Functions with variable number of arguments.

Input/output. Concepts. Stream. Buffering. `stdin`, `stdout`, `stderr`. `*printf`, `*scanf`.

Memory allocation. `malloc()`, `calloc()`, `free()`, `alloca()`.

`setjmp()/longjmp()` - concepts, understanding how they are working, how it can or can not be used.

Coding standard.

Flavours of C: K&R C vs. ANSI C vs. ISO C99.

C99 types: int8_t, int16_t, uint8_t, and so on.

4. 4. Network concepts

OSI 7-layer model (at least know, what does OL logo mean :))

IP protocol concepts: IP addresses, routing principles, MTU, fragmentation and disassembling, usage of ICMP.

Ethernet concepts: MAC addresses, ethertype, collisions, ARP.

Service provided by TCP and UDP. TCP and UDP ports. Client-server interaction model.

Domain name. Domain name resolution.

Berkeley socket API (AF_INET, AF_UNIX).

Network configuration and debugging tools: ifconfig, ip, arp, route, traceroute, ping, tcpdump, host, dig, netcat.

Internet services: ftp, telnet, rsh, ssh, ldap, http, https, nfs (concepts, not protocol/configuration/implementation details).

Documents: RFC 791, 792, 826; info sockets, man pages

5. 5. Operating Systems concepts

Kernel and user context concepts. Context switching (between processes and between user level and kernel level).

Types of scheduling. Process priorities.

Programming of processes in the single address space (use pthread or RTEMS or VxWorks as example): shared data, semaphores, message queues, timers. Deadlocks.

Tools of process interaction: messages, pipes, sockets, shared memory, etc.

Interrupts, interrupt mask, interrupt handlers, exceptions.

Concept of drivers, file systems.

Unix I/O basics. System calls: open, creat, read, write, close, link, mknod, stat, umask, unlink, fcntl, dup, dup2. Interrelation between above system calls and C library streams (fopen, fclose, fread, fwrite, fprintf, fscanf, feof, ...)

6. 6. Documentation

MS Word or equivalent product

7. 7. Internet technology

Web browser. HTTP protocol, HTML markup language fundamentals. Web server. Cookies. Proxy servers. Client and server side scripting concepts.

E-mail. E-mail client. E-mail server. E-mail protocols concepts: SMTP, POP, IMAP. E-mail attachments.

MIME types, their usage in e-mail and http.

Secure data exchange. SSL. Client and server certificates. Certification authorities. Firewall. Intranet vs. Extranet. Passwords. Password selection and handling. VPN.

Search engines. <http://www.google.com>, <http://www.yahoo.com>, <http://www.yandex.ru>. Query language.

8. 8. Programming concepts

Library and modules.

Interpreter. Compiler.

Algorithms: binary search

Data structures: array, list (single-linked, double-linked, queue).

Memory allocation, usage and freeing. Memory leaks. Preventing memory leaks. obstack.

9. 9. Communication and language skills

English knowledge: ability to read and understand of technical literature without vocabulary. Ability to write English technical text and communicate orally in English is very appreciated.

Ability to communicate by e-mail in English.