**1. 1.1**

**- important directories -Linux Basic, kernel , shell, Files,**

**- commands - COMM, permission, chmod,**

**- unix permission - changing file attributes, ln(create a link), touch**

**- Emacs 1.2**

**- intro to emacs - man, sticky bit, sticky bit, setuid, setgid, Emacs, ps, kill - daemon,**

**2. 2.1**

**- Unix wildcards and basic regular expressions - Locale, sort, comm, and tr,**

**- some advanced commands - shell scripting, if/loops, stand streams**

**- pipping and redirection - pipline, redirection, regular expression,POSIX**

**- interpreted language - grep, sed,**

**- shell scripting 2.2 - more grep, expression, sample code script,**

**- hard link check**

**3. 3.1**

**- Make - file decompress, compilation process, compilation,**

**- Automake and autoconf - patching, diff,**

**- Cmake 3.2 - python,**

**- patch**

**4. 5.1 - Debugging, GDB, process memory layout, stack info,**

**- C language and its difference with other program, compiling using gcc,**

**- Formatted I/O, stdin, std out, putchar, getchar**

**2. 5.2 - C language, Pnt to function, qsort, struct,**

**- Debuggers - Dynamic memory(malloc, realloc),**

**- debugging tools - readin/ writing (getchar/ putchar), formatted I/O**

**- GDB -**

**- valgrind**

**- strace**

**5. 6.1 - Mode, kernel Mode, memory protection, system calls**

**- Buffers , Buffered I/O - library function, unbuffered vs buffered I/O, time**

**- Why do we want to use buffers - strace,**

**- Buffer overruns and techniques for avoiding them**

**6.2 - read/ write system calls, fstate system calls,**

**2.**

**- system calls vs. library calls**

**- how to use systems calls**

**- c ad system programming**

**6. 7.1**

**- Open CL - multithreading, parallelism, thread, multitasking vs multithreading**

**- Open MP - memory layout(single or multithread program), shared memory**

**- Massive parallelism 7.2 - Ray tracing, pthread function, pthread parameter**

**- clusters, grids, clouds - pthread join**

**2.**

**- Thread vs processes**

**- Multithread vs processes**

**- Multithread performance**

**- POSIX Threads**

**- Thread synchronization**

**7.**

**- Security threads**

**- - Authentication and authorization**

**- chains of trust**

**- firewalls and sandboxes**

**- intrusion detection**

**- backups**

**- security policies**

**2.**

**- Open SSH**

**- SSH-agent**

**- GNU privacy guards**

**- public and private key**

**8.1 - Building and Executable file, static linking,**

**- Dynamic linking, linking and loading, advantage dynamic linking**

**8. - Disadvantage Dynamic linking,**

**- Dynamic linking 8.2 - GCC flags, attributes and function,**

**- from source file to executable**

**- static linking**

**- Dynamic linking**

**- Two ways of doing dynamic linking**

**- why dynamic linking**

**- dynamic linking practice**

**- How to do dynamic linking in makefile**

**9. 4.1 - Types of Version control, git states, command, git merge**

**- Vestion control - git rebase,**

**- CVS, GIT, Subversion 4.2 - Git branch,**

**- Details of GIT**