☐ Q1

△ Using signal, we can create a signal call "SIGCLD" and used in child process, when child raise this signal, parent will catch it by using signal handle and invoke wait( ).

X Q2:

egrep "[0-f]{6}" <file> 0x[0-9A-Fa-f]{6}
or
[0-9A-Fa-f]{6}

☐ Q3:

△ The server program uses a well-known port number so that it can be accessed, reached by any client who initiates its connection. The client use emperical port number because client request packet contains ports such that the server can reach it back throught it.

✓ Q4. salary = 2000
[[$salary -ge 1000]] && echo I have good salary

Q5 command substitution is that the command can substitute itself by the output of this command
e.g. file ($which is) ⇒ file $(which is)
file '$which is' ⇒ file 'which is'
(here, "which is" command gives an output and that output will replace "which is" for "file" command.

✓ Q6. int n_x = htonl(x)

✓ Q7. critical section is a piece of code that only one thread can access at each time
人
access the shared variable and

Q8. egrep "foo | bar" test.txt

Q9. profiling can measure the program's performance and behavior it can tell the programmer where to optimize

Ans: it is process to measure the performance behavior of a program, such as time cost, memory usage of instructions, modules ....

Q10. folk( )
or pthread_create( )

Q11. Application
transportation
network
linking Data link
physical physical layers

Q12. ① Using Atomics, make sure each write to the same location is seen by the same order by all threads
② Using Locks ③ mutual exclusion

Q13. Used to find ip address, since ip address is just bunch of numbers DNS is easier for us to understand.
Ans: find IP address given a domain name

Q14. $T = \frac{1}{0.2 + \frac{0.8}{4}} = \frac{1}{0.2+0.2} = \frac{1}{0.4} = \frac{5}{2}$
$= 2.5\times$

Q15. backlog is the number of requests in queue.
if backlog set to 0 which's that you connect to the server but client does not request any thing. so the client request might get drop when server is busy

Q16.
```
int cur_write = 0;
while (cur_write < 400) {
    cur_write += write(100, buf + cur_write, 400 - cur_write)
}
```

Q17:
soft quote will allow variables to use
hard quote will quote everything inside '...'.
say set Var = 2
soft quote: "$Var" it will print out 2
but hard quote: '$Var' it will print out $Var

Q18:
socket is the end point of connection.
it is uniquely identified by ip address and port.

Q19:
```
#! /bin/bash
Var = 1
while ["$Var" -lt 99999]; do
    echo "$Var"
    Var = "1 $Var"
done
```

Socket1: is the passive socket used by the server to receive "initial" request
Socket2: is the active socket which is used later to communicate with the client to exchange the following message (through read/write).

Q20: accept( ) will generate new socket for client, it will be used for read and write later on.

Q21: PWD: the current directory
USER: the name of current user
HOME: home of current user
PATH: where to search executable file
PS1: basic prompt

directory