西南林業大學
SOUTHWEST FORESTRY UNIVERSITY

# 本科生实习报告

实习名称 《计算机网络》课程实习

学　　院 大数据与智能工程学院

专　　业 计算机科学与技术

年　　级 2023

指导教师 王晓林

姓　　名

学　　号

2024 年 12 月 21 日

西南林业大学教务处

# 实习环节情况说明

1. 实习应拟定实习指导书或计划，实习报告应规定基本内容，拟定框架要求。
2. 实习期间，学生应每日填写实习日志，内容包括：实习时间、地点、单位、内容、收获和体会，也可摘抄实习实 测数据资料。实习结束后按照要求认真撰写实习报告。实习日志和实习报告在实习完成后一起交指导教师，作为实 习成绩评定的依据。
3. 指导教师按照实习计划的要求，根据学生的实习日志、实习报告、纪律、表现等综合评定实习成绩。

凡有下列情况之一者实习成绩为不合格：

（1）未达到实习计划规定的基本要求；

（2）实习报告混乱，分析有原则性错误；实习日志缺失，或内容大部分未完成；

（3）实习缺课三分之一以上或无故旷课三天以上；

（4）实习中严重违反纪律。

实习天数：__2__ 实习时间：__2024/12/16 - 2024/12/17__
实习单位及地点：__校本部生物楼 219 机房__

# 目录

# 1 实习的任务、作用和目的

## 1.1 实习目的

掌握 Linux 平台上常用网络工具的使用，并理解基本的网络编程过程。

1. 掌握用 tcpdump 捕获并分析数据包
2. 掌握用 netcat 完成简单的网络会话
3. 掌握基本的网络编程

## 1.2 实验要求

1. 在 Linux 平台完成所有实验
2. 在 Linux 平台完成实验报告
3. 努力尝试用英文撰写实验报告
4. 将实验作业及报告以 tgz 格式打包，并上传到指定教学网站
5. 迟交报告将被扣分

# 2 实习主要内容（含工具、方法等）

详见《实习指导书》。

- https://cs6.swfu.edu.cn/~wx672/lecture_notes/network_basics/proj-week/proj-week.html

# 3 Packet analysis

Upon running the following command:

```
sudo tcpdump -ilo -nnvvvxXKS -s0 port 3333
```

the following packet is captured:

```
08:34:10.790666 IP (tos 0x0, ttl 64, id 12824, offset 0, flags [DF],
proto TCP (6), length 64)
  127.0.0.1.46668 > 127.0.0.1.3333: Flags [P.], seq 2400005725:2400005737,
ack 373279396,
  win 512, options [nop,nop,TS val 3259949783 ecr 3259896343], length 12
      0x0000:  4500 0040 3218 4000 4006 0a9e 7f00 0001  E..@2.@.@.......
      0x0010:  7f00 0001 b64c 0d05 8f0d 2e5d 163f caa4  .....L.....].?..
      0x0020:  8018 0200 fe34 0000 0101 080a c24e e2d7  .....4.......N..
      0x0030:  c24e 1217 6865 6c6c 6f20 776f 726c 640a  .N..hello.world.
```

## 3.1 Command line

Tell me the meaning of each option used in the previous command.

**-i:** Specify the network interface card to listen from.

**-nn:** Don't print addresses.

**-vvv:** Verbose output.

**-x:** When parsing and printing, in addition to printing the headers of each packet, print the data of each packet, including its link level header, in hex.

**-X:** When parsing and printing, in addition to printing the headers of each packet, print the data of each packet (minus its link level header) in hex and ASCII. This is very handy for analysing new protocols. In the current implementation this flag may have the same effect as -XX if the packet is truncated.

**-S:** Print absolute, rather than relative, TCP sequence numbers.

**-K:** Don't attempt to verify IP, TCP, or UDP checksums. This is useful for interfaces that perform some or all of those checksum calculation in hardware; otherwise, all outgoing TCP checksums will be flagged as bad.

**-s0:** Setting snaplen to 0 sets it to the default of 262144, for backwards compatibility with recent older versions of tcpdump.

### 3.2 Packet analysis

Please analyze this captured packet and explain it to me as detailed as you can.

**Answer:** 4 means IPv4; 5 is the header length in word; The 00 following 45 are the ToS bits; 0040 is the total length of this IPv4 packet; 3218 is the identification field; 4000 are the 2 flag bits plus the fragment offset field; 4006 means TTL is 0x40, 0x06 means TCP; 0a9e is the header checksum; 7f00 0001 is the IPv4 address of the loopback interface; b64c is the source port number; 0d05 is the destination port number, 3333 in decimal;

## 4 HTTP

### 4.1 Write a simple script showing how HTTP works (you need `curl`)

### 4.2 Record your HTTP demo session with `ttyrec`

1. Open a new terminal window without tmux.
2. Start `ttyrec`.
3. Start `tmux`.
4. Run `http-demo.sh`.

```
#!/bin/bash
tmux rename-window "HTTP demo"

tmux split-window -vfl75%
tmux split-window -vl12

tmux send-keys -t{top} "curl http://cs6.swfu.edu.cn"
tmux send-keys -t{up-of} "watch -tn.1 'ss -4ant dst :http'" C-m
tmux send-keys "sudo tcpdump -i wlp1s0 port http" C-m

tmux select-pane -t{top}
```

程序 4-1: http-demo.sh

# 5 Socket programming

## 5.1 TCP

```
#include <sys/un.h>
#include <sys/socket.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define SV_SOCK_PATH "/tmp/us_xfr"
#define BUF_SIZE 100
#define BACKLOG 5

int main(void)
{
  struct sockaddr_un addr;
  int sfd, cfd;
  ssize_t numRead;
  char buf[BUF_SIZE];

  if( (sfd = socket(AF_UNIX, SOCK_STREAM, 0)) == -1 ){
      perror("socket");
          exit(EXIT_FAILURE);
  }

  memset(&addr, 0, sizeof(struct sockaddr_un));
  addr.sun_family = AF_UNIX;
  strncpy(addr.sun_path, SV_SOCK_PATH, sizeof(addr.sun_path) - 1);

  if( bind(sfd, (struct sockaddr *)&addr, sizeof(struct sockaddr_un)) == -1
  ↪  ){
      perror("bind");
          exit(EXIT_FAILURE);
  }

  if(listen(sfd, BACKLOG) == -1){
```

```
        perror("listen");
            exit(EXIT_FAILURE);
    }

    for(;;) {
        if( (cfd = accept(sfd, NULL, NULL)) == -1 ){
                    perror("accept");
                    exit(EXIT_FAILURE);
                }

        while((numRead = read(cfd, buf, BUF_SIZE)) > 0)
            if(write(STDOUT_FILENO, buf, numRead) != numRead){
                perror("partial/failed write");
                            exit(EXIT_FAILURE);
                        }

        if(numRead == -1){
                    perror("read");
                    exit(EXIT_FAILURE);
            }

        if(close(cfd) == -1){
                    perror("close");
                    exit(EXIT_FAILURE);
            }
    }
}

```

程序 5-2: A simple TCP server

```
#include <sys/un.h>
#include <sys/socket.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define SV_SOCK_PATH "/tmp/us_xfr"
#define BUF_SIZE 100

int main(void)
{
    struct sockaddr_un addr;
    int sfd;
    ssize_t numRead;
    char buf[BUF_SIZE];

    if( (sfd = socket(AF_UNIX, SOCK_STREAM, 0)) == -1 ){
        perror("socket");
        exit(EXIT_FAILURE);
    }
```

```
21
22      memset(&addr, 0, sizeof(struct sockaddr_un));
23      addr.sun_family = AF_UNIX;
24      strncpy(addr.sun_path, SV_SOCK_PATH, sizeof(addr.sun_path) - 1);
25
26      if (connect(sfd, (struct sockaddr *) &addr, sizeof(struct sockaddr_un))
        ↪  == -1){
27          perror("connect");
28                      exit(EXIT_FAILURE);
29      }
30
31      while ((numRead = read(STDIN_FILENO, buf, BUF_SIZE)) > 0)
32          if (write(sfd, buf, numRead) != numRead){
33              perror("partial/failed write");
34                              exit(EXIT_FAILURE);
35                      }
36
37      if (numRead == -1){
38          perror("read");
39                      exit(EXIT_FAILURE);
40      }
41
42      exit(EXIT_SUCCESS); /* Closes our socket; server sees EOF */
43  }
```

程序 5-3: A simple TCP client

### 5.2 UDP

```
1  /* A simple UDP server written in C */
2
3  // Your code
```

```
1  /* A simple UDP client written in C */
2
3  // Your code
```

## 6 Questions

List at least 5 problems you've met while doing this work. While listing your problems, you have to tell me:

1. Description of this problem. For example,
   - What were you trying to do before seeing this problem?
2. How did you try solving this problem? For example,
   - Did you google? web links?
   - Did you read the man page?
   - Did you ask others for hints?

### 6.1 Problem 1

### 6.2 Problem 2

### 6.3 Problem 3

### 6.4 Problem 4

### 6.5 Problem 5

## 7 实习总结

不要少于 300 字。

## 8 指导教师评语

成绩:

指导教师（签名）:

年　　月　　日