ICS Homework 9

Cao Shengcao

11.8

In the original version of `tiny.c`, the server explicitly waits for a CGI child to terminate by `Wait(NULL)` in `serve_dynamic()`. Now we delete it, and instead let the server reap its children inside a handler.

```
void sigchld_handler(int sig)
{
    int olderrno = errno;
    while (waitpid(-1, NULL, WNOHANG) > 0);
    errno = olderrno;
}
```

Please note the use of `while` because one `SIGCHLD` signal sometimes indicates multiple children termination to be handled. And don't forget to set this handler in `main()` before any child is created.

```
    Signal(SIGCHLD, sigchld_handler);
```

11.9

In the original version of `tiny.c`, the server uses `mmap()` to serve static contents.

```
    /* Send response body to client */
    srcfd = Open(filename, O_RDONLY, 0);    //line:netp:servestatic:open
    srcp = Mmap(0, filesize, PROT_READ, MAP_PRIVATE, srcfd, 0);//line:netp:servestatic:mmap
    Close(srcfd);                           //line:netp:servestatic:close
    Rio_writen(fd, srcp, filesize);         //line:netp:servestatic:write
    Munmap(srcp, filesize);                 //line:netp:servestatic:munmap
```

Now we use `malloc()` instead.

```
    srcfd = Open(filename, O_RDONLY, 0);
    srcp = (char*)Malloc(filesize);
    Rio_readn(srcfd, srcp, filesize);
    Close(srcfd);
    Rio_writen(fd, srcp, filesize);
    free(srcp);
```

Don't forget to `free()` the allocated memory.