Proxy 3-2

담당 교수 : 최상호

교수님

분반 : 목요일 7, 8교시

학번 : 2018202060

전공 :컴퓨터정보공학부

이름 : 이준형

Contents

1.Introduction : 과제 내용에 대한 설명

2.Flowchart : 설계한 과제의 플로우차트를 그리고 설명

3.Pseudo code : 과제에서 사용한 알고리즘의 동작을 설명

4.결과화면 : 모든 명령어에 대해 결과화면을 캡처하고 동작을 설명

5.고찰 : 고찰 작성

6.Reference : 참고한내용

Introduction

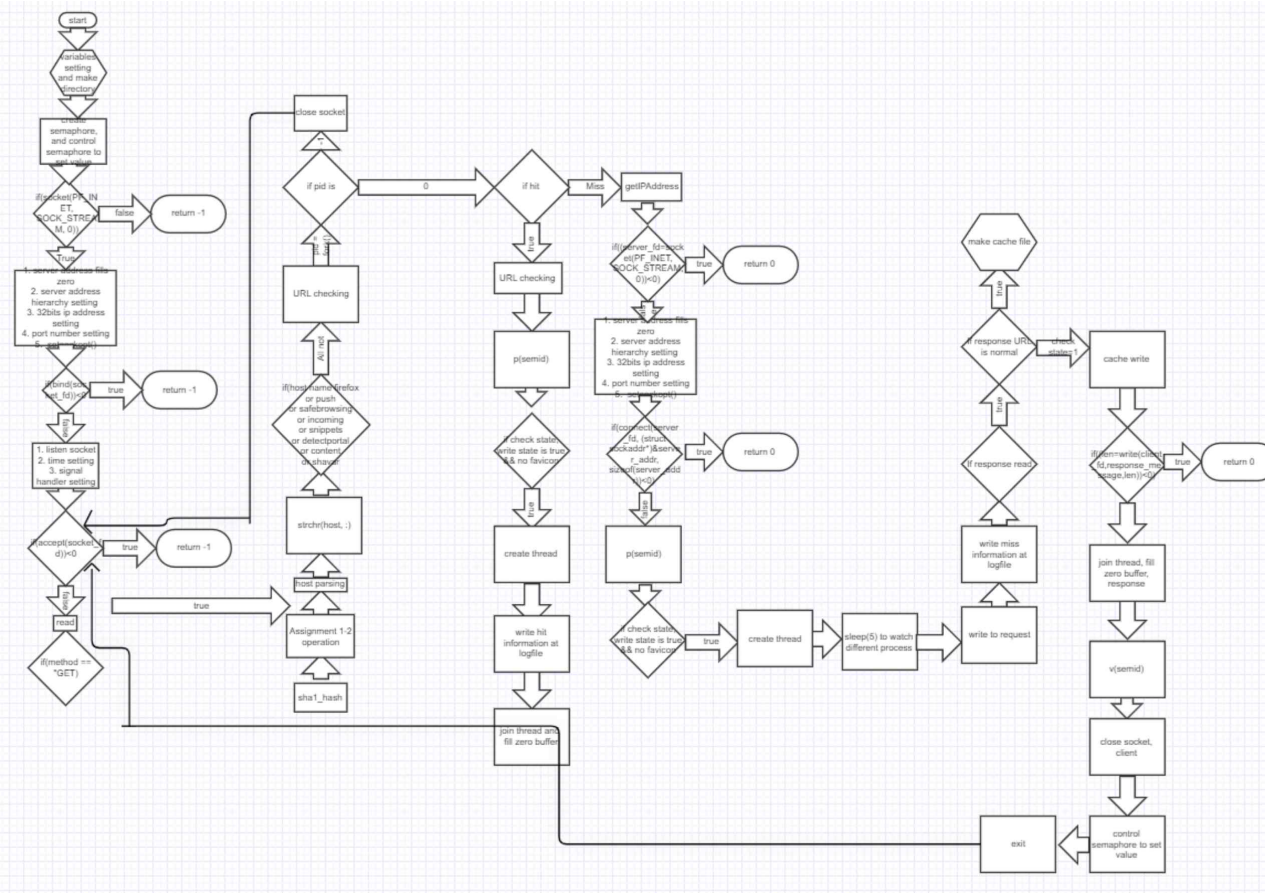
과제에 내용에 대한 설명

이번 시스템프로그래밍 3-2 과제는 기존에 SHA-1 알고리즘함수와 socket함수와 signal들과 critical selection들 중에서 semaphore 그리고 thread를 사용해서 구현했던 proxy_cache를 수정해서 server에서는 firefox로 부터 url을 입력받고 semaphore를 통해 다수 자원을 관리하며 thread를 통해서 프로세스들의 흐름을 나타내면서 2-4와 같은 연산을 하고, MISS이고 critical selection일 때 각 URL을 요청했을 때 이에 관한 모든 Response들을 Cache에 기록을 저장하고 URL을 hashing하여 만든 하나의 파일에모든 Response 기록 한 후 ctrl+c인 SIGINT signal을 받으면 서버 관련 정보를 기록하고 프로그램을 종료시키는것이 이번 과제의 목표입니다.

Flowchart

설계한 과제의 플로우차트를 그리고 설명

제가 구현한 proxy_cache.c의 Flowchart입니다.



먼저 사용할 변수들을 선언하고, semget으로 semaphore을 create해주고, semctl로 control로 semaphore의 value를 set합니다. Pthread_create를 통해 thread가 정상으로 create되는지 확인합니다. 문자열들을 할당해주고, logfile의 directory와 logfile.txt를 만들어줍니다. socket함수를 통해 socket통신을 시작하고, socket에서 사용할 server address들을 세팅 해줍니다. 그 후, bind를 사용하고 firefox로부터 url을 받기위해 listen을 하고, 시작time을 설정하고 signal을 handler들을 설정합니다. 그 후 while문에서 accept를 한 후 read를 통해 method가 GET인지 확인합니다. GET인 경우 sha1_hash함수를 통해서 1-2과제에서 했던 연산을 수행하고, host를 parsing한 후, host의 이름에서 main URL인 경우 URL에 png가 섞여있는지 확인을 하고 fork를 합니다. fork값이 음수인 경우는 socket을 닫습니다. 0(자식프로세스)인경우는 hit인지 miss인지 받은 값을 바탕으로 HIT인 경우, wait함수와 비슷한 역할을 한다고 하셨던 p함수를 사용했고, 확인이 되었는지, favicon URL인지, write를 할 수 있는 상태인지 확인 한 후 PID가 TID를 만들었다고 출력을합니다. hit information을 write합니다. 그 후 다른 process들을 확인하기 위해 sleep함수를 사용하고, sleep이 완료되면 TID가 종료되었다고 출력합니다. miss인 경우 getIPaddress함수에 있는 gethostbyname을 통해서 IP address를 받은 후, HIT인 경우, 확인이 되었는지, favicon URL인지, write를 할 수 있는 상태인지 확인합니다. 이것도 마찬가지로 PID가 TID를 만들었다고

출력을합니다. 그 후 request를 write한 후 다시한번 wait함수와 비슷한 기능을 하는 p함수를 사용합니다. miss information을 logfile에 write합니다. 그리고 다른 process들을 확인하기 위해 sleep을 사용했고, sleep이 완료되면 TID가 종료되었다고 출력합니다. 그 후 URL의 response를 통해 cache file을 만들고 cache에 write합니다. 그 후에 signal함수와 비슷한 기능을 하는 v함수를 사용합니다. 그 후 waitpid와 비슷한 pthread_join을 통해서 thread가 종료되기를 기다리고 thread의 종료코드를 저장합니다. 그리고 다시 semctl함수를 사용해서 다시 한번 semaphore의 값을 초기화 하고 위와 같은 1-2과제 연산을 반복합니다. SIGINT인 ctrl+c를 입력받을때 까지 이와 같은 동작을 수행합니다.

Pseudo code

과제에서 사용한 알고리즘의 동작을 설명

0. semaphore을 create하고 control하기
semkey를 port번호(39999)와 맞춘 후, semget한 값이 -1인지 확인
    음수인 경우 에러 출력 및 종료
semctl값이 음수인지 -1인지 확인
    음수인 경우 에러 출력 및 종료

1. cache file directory와 logfile directory만들기
umask(0) 설정
getHomeDir함수에서 getpwuid를 통해 home directory 반환
home directory에 "/cache" 혹은 "/logfile", "logfile.txt" 를 strcat해서 변수 저장
그 변수의 문자열 그대로 0777권한으로 mkdir 혹은 creat

2. proxy server socket하기
socket한 값이 음수인지 확인
    음수인 경우 0반환(프로그램 종료)
server address값 초기화 하기

3. proxy server bind하기
setsockopt를 통해 can't bind가 되는 경우를 방지
bind한 값이 음수인지 확인
    음수인 경우 서버에서 can't bind를 출력한 후, 0반환(프로그램 종료)
listen하기

4. signal들을 handle하기
1) SIGALRM
alarm(t) 진행
    t초가 지났다면 응답없음 출력 후, kill
2)SIGINT
getpid()를 통해 전역변수로 설정한 pid와 맞는지 확인
    pid가 일치한다면 종료 시간을 설정한 후, 종료 정보를 logfile.txt에 기록
3) SIGCHLD
자식프로세스가 종료될때까지 반복문 실행

5. client accept하기
accept한 값이 음수인지 확인
    음수인 경우 문자열 출력 후, 0 반환(프로그램 종료)

6. method 확인하기
accept한 후 read하기
read한 후 " "가 나오면 strtok하기
method가 "GET"인지 확인
    일치 한다면 URL을 strcpy하기

과제 1-2 연산 수행

7. 예외처리하기

host를 확인하기 위해 "Host:"가 나오면 strtok

host가 firefox, push ,safebrowsing ,incoming, snippets, detectportal, content, shavar인지 확인

전부 일치 하지 않는다면 url parsing하기, write 할 준비하기, process count증가하기, url이 정상인지 .png인지 확인하기

8. fork를 통해서 child process만들기

fork를 한 값이 음수인지 확인

socket을 close하고 continue, 0이라면 hit인지 miss인지 판별

hit인 경우 url이 정상인지 확인하기

정상인 경우 response를 read, p(),write하기

URL에 favicon이 있는지, write state가 1인지, 확인이 되었는지 확인

thread 생성하기

thread가 제대로 생성되지 않았으면 error 출력

모든 조건이 만족한다면 logfile에 hit information 작성하기

그 후 write state를 다시 0으로 하기

thread join하기

miss인 경우 getIPAddress 함수에서 gethostbyname을 통해 IPAddress를 반환하기

server에 소켓한 값이 음수인지 확인

음수인경우 다시 돌아가기

받은 IPAddress를 바탕으로 server address값 초기화하기

server address에 connect한 값이 음수인지 확인

음수인경우 다시 돌아가기

request write하기, p()

URL에 favicon이 있는지, write state가 1인지, 확인이 되었는지 확인

thread 생성하기

thread가 제대로 생성되지 않았으면 error 출력

모든 조건이 만족한다면 logfile에 miss information작성하기, 프로세스 카운트 증가, write state다시 0으로 하기

thread join하기

양수라면 cache파일을 만들고, cache파일에 response저장

buffer와 response 0으로 fill

server close하기

v(), client close하기

semctl값이 음수인지 -1인지 확인

음수인 경우 에러 출력 및 종료

exit()을 통해 자식프로세스 종료

결과 화면

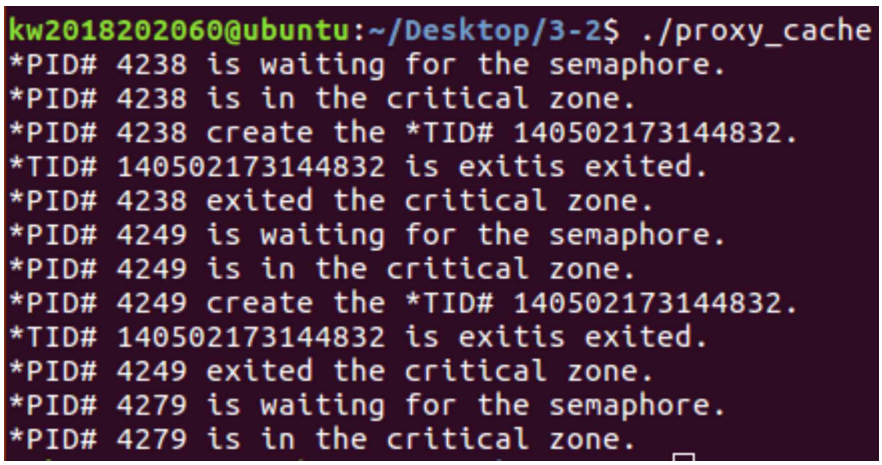
모든 명령어에 대해 결과화면을 캡처하고 동작을 설명

이번 실습에서의 test case를 위해 생각한 시나리오는 다음과 같습니다.

(다른 process들을 보기위해 sleep함수를 사용했습니다.)

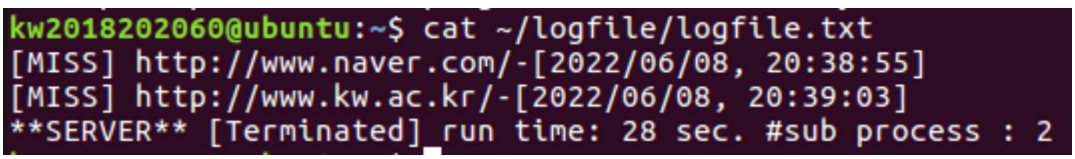
www.naver.com ->(sleep(8), firefox새 탭) -> ~~www.kw.ac.kr.com~~ ->(sleep(8),firefox새 탭)

ctrl+C



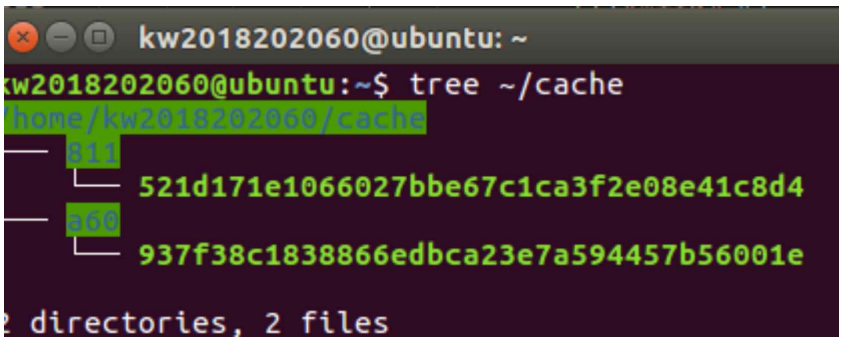
firefox에 입력했을때 출력되는 모습

(마지막 www.daum.net이 입력되지 않은 이유는 고찰에 서술하겠습니다.)



miss가 출력 되는 모습

(4번째 줄의 daum.net가 miss처리가 된 이유는 고찰에 서술하겠습니다.)



cache directory 구조

고찰

고찰 작성

이번 과제는 시스템프로그래밍 중 socket(), signal(), semaphore함수, pthread함수 등 시스템함수들을 사용해서 문제를 해결했습니다. 기존과제보다 조금 더 눈에 보이고 시스템적으로 디테일한 과제를 해결해가면서 점점 더 컴퓨터공학인으로써 성장하는 자신이 뿌듯했습니다.

3-1을 제출하고 시험준비기간이라 빠르게 3-2를 해결하려 했습니다. Thread개념을 처음 접근해서 겁이났지만, 기존 실습을 통해 프로세스를 이해하고 조교님의 thread설명을 들으니 조금이나마 이해를 했습니다. 그리고 조교님께서 pthread_join을 주석처리하면 어떻게 되나도 생각하면서 했는데, 강의 자료에 나와있는것을 주석처리하면 출력문과 같이 입력받을 버퍼에서 같이 나와야하는 '%'가 thr_fn함수를 처리할때 출력문과 같이 나오는것을 확인할 수 있었습니다.

시스템프로그래밍 과목을 통해 시스템함수에 대한 이해도도 가졌지만, server와 client의 개념을 가진 Proxy server를 직접 구현하는 프로젝트를 하면서 평소의 server와 client의 개념도 이해했습니다. 좋은강의와 궁금증을 받아주신 조교님께 감사하다는 말씀 전하며 고찰을 마치겠습니다.