Objective:

응용 프로그램의 성능과 확장성을 향상시키기 위해 운영 체제 커리큘럼 단원에서 습득한 지식을 사용하여 단순화된 HTTP 서버 및 테스트 클라이언트를 개발합니다.

1. Problem definition

HTTP 서버와 클라이언트에서 사용할 수 있는 참조 예제에서 시작하여 학생들이 동시 프로그래밍 패러다임에서 시스템에서 사용 가능한 리소스를 보다 효과적으로 사용하기 위한 메커니즘을 구현하도록 합니다. 시스템 성능 메트릭은 고객의 관점에서 총 응답 시간(HTTP 요청 전송과 응답 수신 사이에 경과된 기간)과 관련됩니다. 이러한 방식으로 서버는 다중 프로세스 구현(1단계) 또는 멀티태스킹(2단계)을 기반으로 하고 클라이언트는 벤치마킹 도구로 진화해야 합니다.

1. Requirements

제출의 첫 번째 단계에서는 다음 요구 사항이 고려됩니다.

A. (20%) 크기 M의 배치에서 인수 N으로 http 요청을 수신하도록 클라이언트 코드를 조정합니다. 각 배치의 프로세스는 동시에 실행되어야 합니다. 프로그램은 이미 배송된 모든 요청에 대한 응답을 받은 후에만 다음 배치로 진행합니다.

a. ./ client 1000 10 -> 1000 requests, 10 to 10

b. ./ client 1000 1000 -> 1000 competing requests

c. ./client 1000 1 -> 1000 sequential requests (serial execution)

프로그램은 M n/m 하위 프로세스 시간을 게시해야 합니다. m개의 자식은 동시 방식으로 실행되어야 하며, 각각은 HTTP 응답 코드(rc) 및 경과 시간(t)을 수집하고 프로세스 식별자(pid), 배치 시퀀스 번호(bsn) 및 배치 컨텍스트에서 시퀀스 번호(rsn)를 요청합니다. 종료하기 전에 각 자식 프로세스는 동일한 공유 파일에 다음 형식의 실행 결과가 포함된 줄을 작성해야 합니다.

pid;bsn;rsn;rc;t

B. (10%) 주문 실행 보고서를 생성합니다. 총 시간, 주문당 평균 시간, 최소 및 최대 시간. 결과는 클라이언트 실행을 완료하기 전에 터미널에 인쇄되어야 합니다.

C. (20%) 요구 사항 A를 변경하여 클라이언트 프로세스가 실행 결과와 공유되는 파일에 줄을 쓰는 대신 파일 쓰기를 담당하는 상위 프로세스(관리자)와 파이프 통신하도록 변경합니다. 파이프의 쓰기 및 읽기는 readn2 e writen2 함수를 사용하여 구현해야 합니다.

도표이(가) 표시된 사진

자동 생성된 설명

D. (15%) 고객의 Graceful Shutdown 기능을 구현합니다. 상위 프로세스는 SIGINT 신호에 대한 서비스 루틴을 설치해야 합니다. 이 신호를 받으면 상위 프로세스는 모든 하위 프로세스를 종료하고 지금까지 수집한 실행 통계(요구사항 B)를 표시해야 합니다.

E. (35%) HTTP 서버를 다중 프로세스 애플리케이션으로 만듭니다.

1. 서버 코드를 조정하여 클라이언트로부터 받은 각 요청을 이행하는 프로세스를 시작합니다("요청 시" 하위 프로세스 생성). 서버가 클라이언트의 연결을 기다리고 있습니다. 새 연결을 수락하면 상위 프로세스가 새 프로세스를 시작하여 요청을 이행합니다. 응답이 전송된 후 자식 프로세스는 연결을 닫고 종료됩니다.

2. 개선된 버전의 요구 사항에는 상위 프로세스가 이전에 릴리스된 프로세스 풀을 유지 관리하는 솔루션 탐색이 포함됩니다. 새 요청이 수신되는 즉시 상위 프로세스는 사용 가능한 하위 프로세스 중 하나로 이를 전달할 책임이 있습니다(waitpid() 시스템 호출 및 지원되는 옵션 참조).

3. 솔루션의 안정성을 보장하기 위해 부모 프로세스는 예기치 않게 종료될 때마다 풀에 삽입하기 위해 새로운 자식 프로세스 생성을 트리거해야 합니다. 예기치 않게 종료된 프로세스의 검증은 신호 응답 루틴(SIGCHLD) 설치를 사용하여 구현되어야 합니다.

1. Notes

프로젝트 전체에서 수강생은 c 라이브러리 함수(fopen, fread, fwrite,...)가 아닌 POSIX 시스템(열기, 읽기, 쓰기 등)에 대한 호출만 사용해야 합니다. 모든 시스템 호출에서 모든 오류 조건은 perror() 함수를 사용하여 테스트해야 합니다.

이 과제는 개별적으로 또는 두 명의 학생이 그룹으로 수행됩니다. 과제물(README 및 소스 코드)은 e러닝 시스템에 표시된 날짜까지 제출해야 하며 교사가 지정한 날짜에 학생이 제시하고 변호합니다. 최종 메모 계산을 위해 각 제출 단계에 50%의 가중치가 할당됩니다.