Oppgave 6 Dokumentasjon

Det startet med at jeg lagde en struct som inneholdt en char array med 11 plasser, bruker array[11] til 0-termineringen
Så lagde jeg en tekstfil.txt fil og skrev noe random der, for å teste om ting funket.
Så lagde jeg en FILE pointer for å lese av filen, som jeg bruker fread() for å lese 10 og 10 bokstaver om gangen for så å plassere de i structen sin char[], Jeg satte også [11] = '\0' med mindre lengden av det som ble lest var mindre enn 10 bokstaver, da satte jeg den på slutten selv med buff[strlen(buff)].

For å veksle mellom main-tråden og arbeids-tråden så valgte jeg å bruke semaphorer, fordi jeg hadde lyst til å prøve det, er ikke sikkert det er den beste løsningen, men det funker helt fint.
Så i main så initialiserte jeg 2 navngitte semaphores med sem_open(), O_CREAT er et flag som jeg bruker I sem_open() og lager semaphoren om den ikke finnes fra før av (include <fcnt1.h> for O_constants). MODE er noe jeg definet(sammen med navnet) og satte til 0644, 0 = prefix til octal, 6 = read/write permisison til user, 4 = read(group), 4 = read(others), dvs “rw-r--r--” det viktigste er at (user) aka oss, har read/write, de andre er ikke så farlig I dette tilfelle(kunne hatt andre verdier) dette blir også brukt til å lage semaphoren.
Til slutt satte jeg valuen til 0 på den ene og 1 på den andre, sånn at jeg kunne bruke sem_post() og sem_wait() for å inkrementere/dekrementere valuen, sånn at de "låste" seg for hver gang en av de skjedde.
så main funksjonen starter med en sem_wait() med den første semaphoren som var 0, og siden jeg starter arbeidstråden sin funksjon med sem_wait() på semaphor2 som var = 1, så blir den = 0(aka arbeidstråd starter først) og avslutter i while-loopen med sem_post() som inkrementerer den valuen, sånn at main kan gjøre sem_wait, noe som da gir et klar-signal til main tråden at det er den sin tur, og sånn veksler jeg imellom de 2 trådene.
Jeg legger også til tallet 10 for hver gang arbeidstråden kjører til en int i structen som da holder styr på hvor mange bokstaver det er totalt. Om filePointern er på slutten av filen(feof) så bruker jeg strlen(buffer)-1 itilfelle bufferet ikke har 10 tegn.
Jeg åpnet også en tråd I main funksjonen som jeg da brukte til å kjøre funksjonen jeg ville at den tråden skulle handle. (pthread_create())
Jeg fant også ut av at jeg ville lagre filnavnet i structen min, og sendte da med hele structen til tråd funksjonen. I structen har jeg også en selvlagd boolean som sier ifra hvis arbeidstråden har nådd feof(), sånn at main tråden også vet når den skal stoppe.
Merk også at hvis det er en newline I tekstfilen, så telles det også som et tegn, når jeg leser av filen.

Til slutt, huske pthread_join() for å sikre at tråden blir ferdig før main, fclose(fp) og sem_close() på begge semaphorene.

Husk at jeg bruker ubuntu, sånn at gcc’en din (I makefilen) blir riktig hvis du skal teste ut
-pthread -lrt (bruker jeg)