

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i / Exam in:	IN2140 - Introduksjon til operativsystemer og datakommunikasjon
Eksamenstid / time of exam:	Vår / Spring 2020
Antall oppgaver / number no tasks:	8

It is important that you read this introduction carefully before you start.	Det er viktig at du leser denne innføringen nøye før du begynner.
General information: <ul style="list-style-type: none">Your submission must be uploaded as a PDF-file in Inspera.Remember that your submission must be anonymous, do not write your name in your submission.All examination support materials are permitted. You need to gather information from available sources, assess the information quality, and put it together in a submission based on your own processing of the content. The submission must reflect your individual level of knowledge.For assignments where it is relevant to use sources and citations, it is important that you do this properly so that you are not suspected of cheating.	Generell informasjon: <ul style="list-style-type: none">Innleveringen din må lastes opp som en PDF-fil i Inspera.Husk at innsendingen din må være anonym, ikke skriv navnet ditt i innleveringen.Alt materiell for eksamensstøtte er tillatt. Du må samle informasjon fra tilgjengelige kilder, vurdere informasjonskvaliteten og sette den sammen i en innsending basert på din egen behandling av innholdet. Innleveringen må gjenspeile ditt individuelle kunnskapsnivå.For oppgaver der det er relevant å bruke kilder og sitater, er det viktig at du gjør dette ordentlig slik at du ikke blir mistenkt for juks.
<ul style="list-style-type: none">You have to read UiO's upload assignment student guide:	<ul style="list-style-type: none">Du må lese UiOs elevguide for opplasting av oppgaver:
https://www.uio.no/studier/eksamen/innlevering/levere-besvarelsen/levere-filopplasting.html	
<ul style="list-style-type: none">You have to read IFI's rules about cheating on exams:	<ul style="list-style-type: none">Du må lese IFIs regler for juks ved eksamen:
https://www.mn.uio.no/ifi/studier/beskjeder/fusk-ved-eksamen-var-2020.html	
How to make a PDF-file:	Hvordan lage en PDF-fil:
https://www.uio.no/tjenester/it/lagring-samarbeid/apne-dokformater/pdf/	

<p>Digital hand drawing:</p> <ul style="list-style-type: none">• If your submission includes digital hand drawings, you are free to use your preferred tools (scanning, cellphone camera etc) as long as everything is readable and delivered as one PDF.• Check out MN's/UiO's recommended solutions for digital hand drawings spring 2020:	<p>Digital håndtegnning:</p> <ul style="list-style-type: none">• Hvis innsendingen inkluderer digitale håndtegnninger, står du fritt til å bruke dine foretrukne verktøy (skanning, mobiltelefonkamera osv.) så lenge alt er lesbart og levert som en PDF.• Sjekk ut MNs/UiOs anbefalte løsninger for digitale håndtegnninger våren 2020:
<p>https://www.uio.no/studier/eksamen/innlevering/alternativer-for-handtegninger.html</p>	
<p>NB! You cannot apply for a postponement of the exam beyond the 7 days the exam is held. If you submit a self-notification about illness you will be able to take a continuation exam.</p>	<p>NB! Du kan ikke søke om utsettelse av eksamen utover de 7 dagene eksamen er avholdt. Hvis du sender inn en egenmelding om sykdom, vil du kunne gå til kotteksamen.</p>
<p>See/Se: https://www.mn.uio.no/om/hms/koronavirus/eksamen-2020.html (§ 9, 10 + 11)</p>	
<p>Contact: User support for exams in the spring of 2020:</p>	<p>Kontakt: Brukerstøtte for eksamer i våren 2020:</p>
<p>https://www.mn.uio.no/om/hms/koronavirus/brukerstotte-og-trosterunde-eksamen-v20.html</p>	

Good luck ! / Lykke til !

Oppgave 1: Function calls vs System calls

English assignment	Norwegian translation
<p>The operating system offers functionality to userspace applications that the application can use to perform operations such as reading from disk or sending packets over the network. This functionality is implemented by system calls. For programmers that write userspace applications, these system calls look identical to function calls that are written by the programmers themselves, or function calls that are supplied by libraries. What are the differences?</p>	<p>Operativsystemet tilbyr funksjonalitet til userspace-applikasjoner som applikasjonen kan bruke til å utføre operasjoner som å lese fra disk eller sende pakker over nettverket. Denne funksjonaliteten implementeres av systemkall. For programmerere som skriver userspace-applikasjoner, ser disse systemskallene ut som funksjonskall som er skrevet av programmererne selv, eller som funksjonskall som leveres av biblioteker. Hva er forskjellene?</p>
<ul style="list-style-type: none">• Describe how function calls work• Describe how system calls work• Explain how they differ	<ul style="list-style-type: none">• Beskriv hvordan funksjonskall fungerer• Beskriv hvordan systemkall fungerer• Forklar hvordan de er forskjellige

Oppgave 2: Process and Threads

English assignment	Norwegian translation
<p>Most modern operating systems allow a userspace application to create new processes and new threads. Sometimes, threads have been called “lightweight processes” because switching between two processes requires more work by the operating system scheduler than switching between threads of the same process.</p>	<p>De fleste moderne operativsystemer lar en userspace-applikasjon lage nye prosesser og nye tråder. Noen ganger har tråder blitt kalt “lette prosesser” fordi det å bytte mellom to prosesser krever mer arbeid av operativsystemets planlegger enn å bytte mellom tråder i samme prosess.</p>
<ul style="list-style-type: none">• Describe what a process is.• Describe what a thread is.• Which resources must be switched between processes that need not be switched for threads?• Does the difference between process and thread matter for scheduling on a server with lots of users and lots of processes like login.ifi.uio.no? Please provide arguments for your view.	<ul style="list-style-type: none">• Beskriv hva en prosess er.• Beskriv hva en tråd er.• Hvilke ressurser må byttes mellom prosesser men ikke for tråder?• Er forskjellen mellom prosess og tråd viktig for planlegging på en server med mange brukere og mange prosesser som login.ifi.uio.no? Argumenter for ditt svar.
<p>Note: In Linux, the best-known system call for creating a new process is fork() and for creating a new thread it is pthread_create().</p>	<p>Merk: I Linux er det mest kjente systemkallet for å opprette en ny prosess fork () og for å opprette en ny tråd er den pthread_create ().</p>

Oppgave 3: Scheduling

English assignment	Norwegian translation
<p>You create an operating system for a monitoring system on an oil rig. About 100 monitoring applications will be running in userspace on your operating system. Each of these applications will receive status updates from a small number of systems on the rig every second. It will use this data to perform complex simulations to predict if a system will fail. When it predicts that a system may fail, it sends a warning message to another computer and continues with its task as if nothing special happened. When its simulation is complete, it waits for a new status update.</p>	<p>Du lager et operativsystem for et monitoreringssystem på en oljerigg. Rundt 100 monitoreringsapplikasjoner kjører i userspace på operativsystemet ditt. Hver av disse applikasjonene vil motta statusoppdateringer fra et lite antall systemer på riggen hvert sekund. Den vil bruke disse dataene til å utføre komplekse simuleringer for å forutsi om en feil vil oppstå i et system. Når den spår at et system kan feile, sender den en advarsel til en annen datamaskin og fortsetter med oppgaven som om ingenting spesielt har skjedd. Når simuleringen er fullført, venter den på en ny statusoppdatering.</p>
<p>The status updates arrive once per second, none of them has priority, no single monitoring application is more important than any other monitoring application. The monitoring system has no user interface, there will never be any input from a human user.</p>	<p>Statusoppdateringene kommer en gang per sekund, ingen av dem har prioritet, ingen monitoreringsapplikasjon er viktigere enn en annen. Monitoreringssystemet har ikke noe brukergrensesnitt, det vil aldri komme noen innspill fra en menneskelig bruker.</p>
<p>How do you design the scheduler for this operating system?</p>	<p>Hvordan designer du scheduleren for dette operativsystemet?</p>
<ul style="list-style-type: none">• Explain preemptive and non-preemptive scheduling.• Discuss whether preemptive or non-preemptive scheduling is appropriate for this system.• Explain for 4 scheduling algorithms how they work and discuss how well each of them is suited for this task.• Propose one of these 4 scheduling algorithms and explain your choice.	<ul style="list-style-type: none">• Forklar preemptiv og ikke-preemptiv scheduling.• Drøft om preemptiv eller ikke-preemptiv scheduling er passende for dette systemet.• Forklar for 4 scheduleringsalgoritmer hvordan de fungerer og diskuter hvor godt hver av dem er egnet for denne oppgaven.• Foreslå en av disse 4 scheduleringsalgoritmene og forklar valget ditt.

Oppgave 4: Virtual memory

English assignment	Norwegian translation
<p>Virtual memory is used by operating systems to allow userspace processes to use amounts of memory that exceed the physical RAM of the computer. Pages of memory that belong to a process can either be located on disk or in RAM. When they are located in RAM, they usually have a physical address that is different from the address used by the user-space process they belong to. Page tables are used to implement the translation from a logical address that the user-space processes use to physical addresses in RAM, as well as for finding out if a page is currently loaded into RAM.</p> <ul style="list-style-type: none"> • Your processor has 32 bit addresses • You have a 4GB address space • The page size of your operating system is 1 kilobyte (1024 bytes) • Every byte can be addressed • You are using multi-level paging to allow processes to be up to 4GB large 	<p>Virtuelt minne brukes av operativsystemer for å tillate at userspace-prosesser bruker mer minne enn datamaskinen har fysisk RAM. Minnesider som tilhører en prosess, kan enten være plassert på disk eller i RAM. Når de er plassert i RAM, har de som oftest en fysisk adresse som er ulik adressen som brukes av userspace-prosessen de tilhører. Page tables brukes til å oversette fra en logisk adresse som userspace-prosessen bruker til fysiske adresser i RAM, samt for å finne ut om en side for øyeblikket er lastet inn i RAM.</p> <ul style="list-style-type: none"> • Prosessoren din har 32-bit adresser • Du har et adresseringsrom på 4 GB • Sidestørrelsen på operativsystemet ditt er 1 kilobyte (1024 byte) • Hver byte kan adresseres • Du bruker multi-level paging for å la prosesser være opptil 4 GB store
Task	Oppgave
<ul style="list-style-type: none"> • Explain (in general terms) how virtual addresses are translated to physical addresses when multi-level paging is used. The explanation must include issues such as memory lookup, page caches, loading pages, and dirty pages. An explanation of page replacement strategies is not expected. • With the given page size, how many levels must your virtual address translation at least have? Explain why that is the case. Don't forget that page tables and page maps are also limited by the 1 kilobyte page size. 	<ul style="list-style-type: none"> • Forklar (generelt sett) hvordan virtuelle adresser blir oversatt til fysiske adresser med multi-level paging. Forklaringen må inkludere problemer som minneoppslag, page cache, innlasting av sider og begrepet "dirty page". Det er ikke nødvendig å forklare strategier for utskiftning av sider. • Hvor mange nivåer må din virtuelle adresseoversettelse minst ha med den oppgitte sidestørrelsen? Forklar hvorfor. Ikke glem at page tables og page maps også er begrenset av 1 kilobyte sidestørrelse.

Oppgave 5: Splitting an address block into networks

English assignment	Norwegian translation
<p>Your company has been assigned 1024 IPv4 addresses starting at address 9.239.16.0. Your company has offices in three different locations, A, B and C, so you must split these addresses into three separate, routable networks, netA, netB and netC. Each of the locations requires 120 public IPv4 addresses right now, but you expect that all of them will need more in the future. You must decide how to assign these addresses.</p>	<p>Firmaet ditt har fått tildelt 1024 IPv4-adresser som starter på adressen 9.239.16.0. Firmaet ditt har kontorer på tre forskjellige steder, A, B og C, så du må dele adressene i tre separate, rutbare nettverk, netA, netB og netC. Hver av de tre stedene har akkurat nå behov for 120 offentlige IPv4-adresser, men du forventer at alle vil trenge mer i fremtiden. Du må velge hvordan du tildeler disse adressene.</p>
<ul style="list-style-type: none">• What is the netmask of the address block that has been assigned to your company?• How many addresses do you allocate to each of the networks, netA, netB and netC? What is the network address and the netmask for each of these networks? Explain how you get to this solution.• Explain why you have chosen to divide your address range into three networks like this. Did you consider future growth in your assignment? Did you keep any addresses in reserve? And if yes, how do you plan to use them when necessary?	<ul style="list-style-type: none">• Hva er nettmasken til adresseblokken som er tildelt selskapet ditt?• Hvor mange adresser tildeler du hvert nettverk, netA, netB og netC? Hva er nettverksadressen og nettmasken for hvert av disse nettverkene? Forklar hvordan du kommer til denne løsningen.• Forklar hvorfor du har valgt å dele adresseområdet ditt i tre nettverk som dette. Vurderte du fremtidig vekst i oppgaven din? Beholdt du noen adresser i reserven? Og hvis ja, hvordan planlegger du å bruke dem?

Oppgave 6: Packets over TCP

English assignment	Norwegian translation
<p>The Transmission Control Protocol (TCP) is a transport protocol that provides connection-oriented, reliable, end-to-end byte-streams as a service to the application layer. For many applications, it is very inconvenient that TCP uses byte streams rather than packets.</p>	<p>Transmission Control Protocol (TCP) er en transportprotokoll som leverer forbindelsesorienterte, pålitelige byte-streams fra ende til ende som en tjeneste til applikasjonslaget. For mange applikasjoner er det veldig upraktisk at TCP bruker bytestrømmer i stedet for pakker.</p>
<ul style="list-style-type: none">• Explain what the statement “TCP provides a byte-stream service” means.• What does the byte-stream service mean for applications that use socket functions (Berkeley sockets) such as send() and recv() to exchange data? What must programmers do differently when they use TCP instead of a transport protocol that provides a packet-based service?• If an application uses TCP but needs packets, how must the developer design the application-layer protocol?• Would it be easy to create a new transport protocol PacketTP with small changes to TCP that delivers a connection-oriented, reliable, end-to-end datagram service? List at least 3 required changes and explain why they are important. The changes can be small.	<ul style="list-style-type: none">• Forklar hva uttalelsen “TCP leverer en byte-stream-tjeneste” betyr.• Hva betyr byte-stream-tjenesten for applikasjoner som bruker socket funksjoner (Berkeley-sockets) som send() og recv() for å utveksle data? Hva må programmerere gjøre annerledes når de bruker TCP i stedet for en transportprotokoll som gir pakkebasert tjeneste?• Hvis en applikasjon bruker TCP men trenger pakker, hvordan må utvikleren designe applikasjonslagsprotokollen?• Er det enkelt å lage en ny transportprotokoll PacketTP med små endringer i TCP, men som leverer en forbindelsesorientert, pålitelig, ende-til-ende datagramtjeneste? Foreslå minst 3 endringer som er nødvendige og forklar hvorfor de er viktige. Endringene kan være små.

Oppgave 7: Flow control

English assignment	Norwegian translation
<p>Flow control on the transport layer does usually solve two problems: it ensures that the sender does not send packets faster than the receiver can receive, and it detects and corrects packet loss.</p> <p>Real transport layers use a credit mechanism instead of the simpler sliding window mechanism with selective repeat.</p>	<p>Flytkontrollen på transportlaget løser vanligvis to problemer: den sikrer at avsenderen ikke sender pakker raskere enn mottakeren kan motta, og den oppdager og retter pakketap. Ekte transportlag bruker en kreditt-mekanisme i stedet for den enklere mekanismen "sliding window with selective repeat".</p>
<ul style="list-style-type: none">● Explain how the flow control mechanism selective repeat works● Explain how the flow control with a credit mechanism works.● Describe the differences between selective repeat and a credit mechanism. Is there anything different about handling packet losses, handling ACK losses or using timeouts? How large can the sequence number space be, and how many of those sequence numbers can be "in flight" at any time?● Can you find a problem with the loss of ACKs in the credit mechanism that does not exist in selective repeat, and do you have a proposal for solving it?	<ul style="list-style-type: none">● Forklar flytkontrollmekanismen "selective repeat".● Forklar flytkontrollen kreditt-mekanisme.● Beskriv forskjellene mellom selective repeat og kreditt-mekanisme. Er det noen forskjell ved håndtering av pakketap, tap av ACKer eller bruk av timeouts? Hvor stort kan sekvensnummerområdet være, og hvor mange av disse sekvensnumrene kan være "in flight" til et gitt tidspunkt?● Kan du finne et problem med tap av ACK-er i kreditt-mekanismen som ikke eksisterer i selective repeat, og har du et løsningsforslag?

Oppgave 8: Building Routing tables from Dijkstra Shortest Path First

English assignment	Norwegian translation
<p>The routing process on Router A in the following network (consisting of nodes A-J) receives distance estimates (number in the graph) for all directly connected nodes in a small network.</p> <p>Router A must now compute the best path to every other node in the network and create its own routing table. The routing process uses Dijkstra's Shortest Path First algorithm for computing distances.</p>	<p>Routing-prosessen på ruter A i følgende nettverk (bestående av nodene A-J) mottar estimerte avstander (nummer i grafen) for alle direkte tilkoblede noder i et lite nettverk.</p> <p>Ruter A må nå beregne de beste stiene til alle andre noder i nettverket og lage sin egen routingtabell. Routing-prosessen bruker Dijkstra's Shortest Path First-algoritme for beregning av avstander.</p>
Do the following:	Gjør følgende:
<ul style="list-style-type: none"> • Explain how Dijkstra's shortest path first algorithm works. You can start with the list of steps that you can find in the lecture slides, online or in a book. It is OK to copy this list. But after that, you must explain every step with your own words as well. Why are these steps important? What do they do? • Use the algorithm to compute the distance from A to every other node. Show at least 3 intermediate 	<ul style="list-style-type: none"> • Forklar hvordan Dijkstras shortest path first algorithm fungerer. Du kan starte med listen over trinn som du kan finne på forelesningsfoilene, online eller i en bok. Det er OK å kopiere denne listen. Men etter det må du i tillegg forklare hvert trinn med dine egne ord. Hvorfor er disse trinnene viktige? Hva gjør de? • Bruk algoritmen til å beregne avstanden fra A til

<p>steps: right after F gets a permanent label, right after E gets a permanent label, right after I gets a permanent label.</p> <p>You choose how to do it, but a corrector must be able to understand it. You could, for example, create tables, or you can draw on a paper and insert a picture in your response.</p> <ul style="list-style-type: none">• Finally, make a table that shows A's routing table, consisting of three columns. First column: destination node, second column: next node on the path to the destination, third column: the length of the shortest path to this one.	<p>alle andre noder. Vis 3 mellomtrinn: rett etter at F har fått en permanent label, rett etter at E har fått en permanent label, rett etter at I har fått en permanent label.</p> <p>Du velger hvordan du skal gjøre det, men en retter må kunne forstå det. Du kan f.eks. lage tabeller, eller du kan tegne på papir og sette inn et bilde i svaret ditt.</p> <ul style="list-style-type: none">• Til slutt lager du en tabell som viser A sin routingtabell, bestående av tre kolonner. Første kolonne: destinasjonsnode, andre kolonne: neste node på stien til destinasjonen, tredje kolonne: lengden på den korteste stien til denne.
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------