

A TANTÁRGY ADATLAPJA

1. A képzési program adatai

1.1 Felsőoktatási intézmény	Babeş-Bolyai Tudományegyetem
1.2 Kar	Matematika és Informatika Kar
1.3 Intézet	Magyar Matematika és Informatika Intézet
1.4 Szakterület	Informatika
1.5 Képzési szint	Alapképzés
1.6 Szak / Képesítés	Informatika – magyar vonal

2. A tantárgy adatai

2.1 A tantárgy neve	Osztott operációs rendszerek						
2.2 Az előadásért felelős tanár neve	dr. Ruff Laura-Ildikó, adjunktus						
2.3 A szemináriumért felelős tanár neve	drd. Varga Levente, tanársegéd						
2.4 Tanulmányi év	II.	2.5 Félév	I.	2.6. Értékelés módja	Vizsga	2.7 Tantárgy típusa	Kötelező

3. Teljes becsült idő (az oktatási tevékenység féléves óraszama)

3.1 Heti óraszám	4	melyből: 3.2 előadás	2	3.3 szeminárium/labor	2
3.4 Tantervben szereplő össz-óraszám	56	melyből: 3.5 előadás	28	3.6 szeminárium/labor	28
A tanulmányi idő elosztása:					óra
A tankönyv, a jegyzet, a szakirodalom vagy saját jegyzetek tanulmányozása					24
Könyvtárban, elektronikus adatbázisokban vagy terepen való további tájékozódás					10
Szemináriumok / laborok, házi feladatok, portofóliók, referátumok, esszék kidolgozása					24
Egyéni készségfejlesztés (tutorálás)					5
Vizsgák					6
Más tevékenységek:					
3.7 Egyéni munka össz-óraszama	69				
3.8 A félév össz-óraszama	125				
3.9 Kreditszám	5				

4. Előfeltételek (ha vannak)

4.1 Tantervi	
4.2 Kompetenciabeli	C/C++ programozási készség, Unix alapú operációs rendszerek ismerete, shell programozás

5. Feltételek (ha vannak)

5.1 Az előadás lebonyolításának feltételei	vetítógép
5.2 A szeminárium / labor lebonyolításának feltételei	saját felhasználói fiók a kar linux szerverén

6. Elsajátítandó jellemző kompetenciák

Szakmai kompetenciák	<ul style="list-style-type: none"> • az operációs rendszerekkel kapcsolatos kérdéskörökben való jártasság: fontosabb összetevők, funkcionalitás, tervezés, különböző algoritmusok, alapelvek ismerete • a mai modern operációs rendszerek felépítésével kapcsolatos kérdéskörök ismerete • folyamatok közti kommunikációs lehetőségek ismerete/használata, többszálúsággal kapcsolatos alapelvek ismerete és ezek gyakorlati alkalmazása, szinkronizációs eszközök ismerete/használata, szinkronizációs problémák megoldása
Transzverzális kompetenciák	<ul style="list-style-type: none"> • mivel az operációs rendszer vezérli gyakorlatilag a különböző felhasználói folyamatok futását, az ennek működésébe való betekintés, a háttérben zajó események mélyebb ismerete segít a hatékonyabb programfejlesztésben. • feladatmegoldó készség, kreatív gondolkodás

7. A tantárgy célkitűzései (az elsajátítandó jellemző kompetenciák alapján)

7.1 A tantárgy általános célkitűzése	<ul style="list-style-type: none"> • megismerkedni az operációs rendszerek általános elméletével (fontosabb összetevők, funkcionalitás, tervezés, különböző algoritmusok, alapelvek) • megismerni a mai modern operációs rendszerek felépítésével kapcsolatos kérdésköröket
7.2 A tantárgy sajátos célkitűzései	<ul style="list-style-type: none"> • a folyamatok közti kommunikáció különböző eszközeinek megismerése • a multitasking-ot illetve többszálúságot (multithreading) támogató operációs rendszerekben fellépő problémák megértése, illetve az ezekkel kapcsolatos megoldási stratégiák elsajátítása • többprocesszoros illetve osztott rendszerek sajátosságai

8. A tantárgy tartalma

8.1 Előadás	Didaktikai módszerek	Megjegyzések
1. hét Bevezetés <ul style="list-style-type: none"> - Operációs rendszerek fejlődésének története - Op. rendszer kérdéskörök - Modern rendszerekhez kapcsolódó fogalmak 	előadás, vetítés, magyarázat, dialógus, konkrét példák keresztül történő szemléltetés, grafikus ábrázolás	
2. hét Folyamatok	előadás, vetítés, grafikus ábrázolás, magyarázat,	

<ul style="list-style-type: none"> - Folyamatok állapotai (jelentésük, állapotmodellek, mi tette szükségessé egy-egy új állapot bevezetését) - Állapotátmenetek - adatsruktúrák, melyekre az op. rendszernek szüksége van a folyamatkezeléshez <ul style="list-style-type: none"> - Memória táblázatok - Ki-/bemeneti eszköz táblázatok - Állománytáblázatok - Folyamattáblázatok - Folyamatkezelés <ul style="list-style-type: none"> - Üzem mód váltás illetve folyamatváltás (melyik miben áll, különbség) 	<p>problémafelvetés, korábbi ismeretek aktivizálása</p>	
<p>3-6. hét</p> <p>Versenyhelyzet</p> <ul style="list-style-type: none"> - Alapfogalmak (kölcsonös kizárás, kritikus szakasz, holtpont, éheztetés) - Megoldandó problémák egymással (különböző mértékben) kölcsönhatásban levő folyamatok esetén <p>Kölcsonös kizárás</p> <ul style="list-style-type: none"> - Kialakulásához szükséges feltételek - Kölcsonös kizárás megvalósítása <ul style="list-style-type: none"> - Szoftver megoldás (Dekker algoritmus, Peterson algoritmus) - Hardver támogatás (megszakítások letiltása, speciális utasítások -Test and Set, Exchange -) - Operációs rendszer támogatás (szemafor, monitor, üzenetek) - tipikus példafeladatok (termelő/fogyasztó, író/olvasó probléma) <p>Holtpont</p> <ul style="list-style-type: none"> - Fellépése újrafelhasználható/elfogyó erőforrások esetén - Holtpont kialakulásának feltételei - Holtpont kezelés: <ul style="list-style-type: none"> - Holtpont megelőzése - deadlock prevention (a holtpont kialakulásához szükséges 4 feltétel 	<p>előadás, vetítés, konkrét példákon keresztül történő szemléltetés, magyarázat, beszélgetés-vita, fakultatív önálló munka</p>	

<p>valamelyikének megszüntetése által)</p> <ul style="list-style-type: none"> - Holtpont kialakulásának kivédése - deadlock avoidance (folyamat kéréseinek megtagadása indulásból, erőforrás-hozzárendelés megtagadása - "bankár" algoritmus -) - Holtpont felismerése és feloldása <ul style="list-style-type: none"> - holtpont felismerésére szolgáló algoritmus - különböző megközelítések a holtpont feloldására - tipikus példa feladat (étkező filozófusok problémája) 		
<p>7. hét</p> <p>Szinkronizáló eszközök a gyakorlatban</p> <ul style="list-style-type: none"> - IPC csomag szinkronizáló eszközei - POSIX szinkronizáló eszközök - Példaprogramok 	<p>gyakorlati példák bemutatása, megbeszélése, dialógus</p>	
<p>8. hét</p> <p>Szálak, SMP architektúra</p> <ul style="list-style-type: none"> - Szálak <ul style="list-style-type: none"> - A fogalom tisztázása (különbségek a folyamathoz képest) - Szálak használatának előnyei - Szálak kezelése (felhasználói- illetve kernelszintű szálak) - Szálak és folyamatok egymáshoz való viszonya - SMP szervezés 	<p>előadás, vetítés, magyarázat, konkrét példákon keresztül történő szemléltetés, grafikus ábrázolás, problémafelvetés</p>	
<p>9. hét</p> <p>Ütemezés</p> <ul style="list-style-type: none"> - Hosszútávú ütemezés - Középtávú ütemezés - Rövidtávú ütemezés <ul style="list-style-type: none"> - osztályozási kritériumok - ütemezési algoritmusok <ul style="list-style-type: none"> - FCFS, round robin, SPN, SRT, HRRN, visszacsatolásos - Fair share (elve) - Ütemezés többprocesszoros rendszerekben 	<p>előadás, vetítés, magyarázat, dialógus, konkrét példákon keresztül történő szemléltetés, grafikus ábrázolás</p>	
<p>10-12. hét</p>	<p>vetítés, előadás,</p>	

<p>Memória-kezelés</p> <ul style="list-style-type: none"> - Memória-kezeléssel kapcsolatos követelmények - Memória-kezelési technikák: <ul style="list-style-type: none"> - Fix particionálás <ul style="list-style-type: none"> - azonos/változó méretű partíciók - előnye/hátránya - Dinamikus particionálás <ul style="list-style-type: none"> - előnye/hátránya - memóriablokk elhelyezése (best-fit, first-fit, next-fit algoritmus alapján) - „Buddy” rendszer - Egyszerű lapozás - Egyszerű szegmentálás <p>Virtuális memória kezelése</p> <ul style="list-style-type: none"> - virtuális memória használatának előnye, alapfogalmak - lapozást támogató hardver - lapozás - szegmentálás - lapozás + szegmentálás - Operációs rendszer stratégiák <ul style="list-style-type: none"> - betöltési stratégia (fetch policy) - elhelyezés (placement) - lapcsere (replacement) <ul style="list-style-type: none"> - lapcsere algoritmusok (optimális, a legrégebben használt, FIFO, clock) - lap-pufferelés - rezidens halmaz kezelése - kiírási stratégia (cleaning policy) 	<p>magyarázat, problémafelvetés</p>	
<p>13. hét</p> <p>Ki-/bemenet kezelés</p> <ul style="list-style-type: none"> - ki-/bemeneti eszközök osztályozása - ki-/bemeneti technikák (programozott I/O, megszakításvezérelt I/O, DMA) - a ki-/bemenet kezelő funkció fejlődése - DMA (szerkezete, működése, szerepe, különböző konfigurációk) - tervezési szempontok - logikai szerkezet 	<p>előadás, vetítés, magyarázat, grafikus ábrázolás</p>	

<ul style="list-style-type: none"> - I/O puffereles - lemez teljesítménye - lemez ütemezése <ul style="list-style-type: none"> - ütemezési algoritmusok (FIFO, LIFO, SSTF, SCAN, CSCAN) 		
14. hét Osztott rendszerek <ul style="list-style-type: none"> - kliens-szerver kapcsolat - osztott üzenetküldés - távoli eljárás hívás 	előadás, vetítés, magyarázat, grafikus ábrázolás, dialógus	
Könyvészet 1. William Stallings, Operating Systems: Internals and Design Principles, 8th ed., Prentice Hall, 2014 http://williamstallings.com/OperatingSystems/ 2. A Tannenbaum, Modern Operating Systems 3rd Ed., Prentice Hall 2007 (4th ed., 2014) 3. Silberschatz, P. Galvin and G. Gagne, Operating System Concepts, John Wiley & Sons, Inc., 2002 (9th ed., 2012) 4. BOIAN F, VANCEA A., BOIAN R., BUFNEA D., STERCA A., COBARZAN C., COJOCAR D., Sisteme de operare Ed. Risoprint, 2006		
8.2 Szeminárium / Labor	Didaktikai módszerek	Megjegyzések
1. rendszerhívások	házi feladatok megbeszélése/ bemutatása/hibajavítás, fakultatív pluszfeladatok	
2. jelzések (signals)		
3-4. osztott memória - szinkronizálás nélkül		
5-6. szemaforok és osztott memória		
7-8. üzenetsorok		
9. POSIX/Solaris szálak (threads) - szinkronizálás nélkül		
10-11. POSIX/Solaris szálak (threads) – szinkronizálással		
12-13. Ütemezés		
14. labortevékenység lezárása		
Könyvészet Buzogány László, Folyamatok, http://www.cs.ubbcluj.ro/~laura/so2/dokumentacio/ Programming in C UNIX System Calls and Subroutines using C, http://www.cs.cf.ac.uk/Dave/C/		

9. Az episztemikus közösségek képviselői, a szakmai egyesületek és a szakterület reprezentatív munkáltatói elvárásainak összhangba hozása a tantárgy tartalmával.

- A tantárgy tartalma megegyezik az egyetemi oktatásban a fontosabb egyetemeken oktatott, operációs rendszerek általános elméletével foglalkozó tantárgyak tartalmával.

10. Értékelés

Tevékenység típusa	10.1 Értékelési kritériumok	10.2 Értékelési módszerek	10.3 Aránya a végső jegyben
10.4 Előadás	Válasz helyessége	Rövid teszt kurzusonként (F)	20%
	Válasz helyessége (alapfogalmak ismerete,	Írásbeli vizsga (V)	30%

	tanult algoritmusok alkalmazásának készsége alapján)		
10.5 Szeminárium / Labor	Program helyessége, határidő betartása	Labor-házifeladatok (L)	30%
	Labor idején megoldott, helyesen működő program, bittologatók előadás/részvétel	Pluszpontok (fakultatív pluszfeladatok, bittologatók) (P+B)	
	helyesen működő program	Gyakorlati vizsga (G)	20%

10.6 A teljesítmény minimumkövetelményei

- Laborvizsgán elért 10 pont (konkrétan megadott alapfunkcionalitások implementálása) a lehetséges 20-ból
 - Írásbeli vizsgán elért 15 pont a lehetséges 30-ból
 - Nincs 2-nél több indokolatlan, laborórákról való hiányzás
- Vizsgajegy ≥ 5 , ahol Vizsgajegy = $(\min(10, \text{round}((E-50+V+G)/10)))$
 $E=F+L+P+B$

Kitöltés dátuma

2017.04.22.

Előadás felelőse

Ruff Laura-Ildikó

Szeminárium felelőse

Varga Levente

Az intézeti jóváhagyás dátuma

.....

Intézetigazgató

András szilárd