

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ
МИНИСТРЛІГІ
ҚАРАҒАНДЫ МЕМЛЕКЕТТІК ТЕХНИКАЛЫҚ УНИВЕРСИТЕТІ

Бекітемін
Бірінші проректор

_____ **А.З.Исағұлов**

«_____» _____ 2008ж.

ОҚЫТУШЫ ПӘНІНІҢ ОҚУ-ӘДІСТЕМЕЛІК КЕШЕНІ

«Нақты уақыттың операциялық жүйелері» пәні бойынша

050702 «Автоматтандыру және басқару» мамандық студенттеріне арналған

Электрмеханикалық факультет

В.Ф.Бырька атындағы өндірістік процестерді автоматтандыру кафедрасы

Кіріспе

Оқытушы пәнінің оқу-әдістемелік кешені:

ӨПА кафедра аға-оқытушысы *Паршина Галина Ивановна*мен

ӨПА кафедра аға оқытушысы *Телбаева Шынар Зарыкбековна*мен даярланды..

_____ӨПА_____кафедра мәжілісінде талқыланды
(кафедра атауы)

Хаттама № 20 " 24 " 06 2008 ж.

ӨПА кафедра меңгерушісі

проф., т.ғ.д. _____ Брейдо И.В. «_____» _____ 2008ж.
(қолы)

_____ЭМФ_____факультетінің әдістемелік бюросымен мақұлданды
(факультет атауы)

Хаттама № 10 " 28 " 06 2008 ж.

Төраға _____ Үмбеталин Т.С. «_____» _____ 2008ж.
(қолы)

1 Оқу жұмыс бағдарламасы

1.1 Оқытушы жайлы мәлімет және байланыс ақпараттары

Паршина Галина Ивановна, ҚарМТУ ӨПА кафедра аға-оқытушы.

Телбаева Шынар Зарыкбековна ҚарМТУ ӨПА кафедра аға-оқытушы.

В.Ф.Бырька атындағы ӨПА кафедра ҚарМТУ бас корпусында, 131 аудиторияда орналасқан, байланыс телефоны 56-51-84 (кафедра), 56-53-25 (4 корпус 107 ауд.), оқытушы электрондық мекені: galin@kstu.kz

1.2 Пәннің сыйымдылығы

Семестр	Кредит саны	Сабақ түрі					СДЖ сағатының саны	Барлық сағат саны	Бақылау түрі
		байланыс сағаттар саны			СОДЖ сағатының саны	барлық сағат саны			
		дәрістер	тәжірибиелік сабақтар	зертханалық сабақтар					
5	2	15	-	15	30	60	30	90	емтихан

1.3 Пәннің сипаттамасы

«Нақты уақыттың операциялық жүйелері» пәні 050702 "Автоматтандыру және басқару" мамандық студенттері үшін негізгі пәндердің бірі болып табылады, мамандық оқу жоспарына сәйкес компонент саны бойынша таңдау кіргізілген.

1.4 Пәннің мақсаты

«Нақты уақыттың операциялық жүйелері» пәнін оқыту мақсатына даму тарихы, НУОЖ-нің топталуы (классификациясы) және НУОЖ-нің негізгі ерекшеліктері, қазіргі НУОЖ –нің құрылу ерекшеліктері және осы білімді НУОЖ басқармасымен жұмыс істейтін әр түрлі мақсаттағы қосымшаларды жобалау кезінде қолдануды оқыту жатады.

1.5 Пәннің міндеті

Пәннің міндеті: маманда қатаң негізгі білімді, жоғарғы математикалық мәдениетті және тәжірибиелік бейімділікті, өндірістік қызметті сәтті жүргізуге жеткілікті және қажетті жаңа білімді өзіндік игеруге мүмкіндік беретін НУОЖ басқармасымен жұмыс істейтін қосымшаларды жобалау және инженерлік есептерді есептеуоблысындағы жетістіктерді қалыптастыру жатады.

Берілген пәнді оқу нәтижесінде студент келесіні игеру керек:

Мағұлмат алу: Нақты уақыттағы ҚЕЭМ-да технологиялық комплекстерді автоматизациялау жүйесін жобалау кезінде әр түрлі операциялық жүйенің функциональды мүмкіндіктері мен критерияларын (өлшемдерін) таңдау, технологиялық процестерді автоматтандыру жүйесін жобалау үшін арналған программалау тілінің функциональды мүмкіндіктері туралы

Білу керек: Нақты уақыттағы операциялық жүйенің құрылымын, негізгі құрылу құрылым принциптерін, қолдану облысын, сонымен қатар мамандандырылған программалау тілінің басты мүмкіндіктері.

Істей білу: Нақты уақыттағы жүйелер мен процестер үшін нақты уақыттағы жүйелердегі қолданбалы есептерді программалау

Тәжірибиелік бейімділікті жию: Қазіргі заманғы НУОЖ мен программалау тілі базасындағы технологиялық кешендерді басқару және бақылау жүйесін жобалау есептерін шешу үшін

1.6 Бастапқы реквизиттер

Берілген пәнді оқу үшін келесі пәндерді меңгеру керек(бөлімдерді(тақырыптарды)нұсқаумен):

Пән	Бөлімдердің(тақырыптардың)атауы
1. Информатика	ПЭВМ архитектурасы, Windows операциялық жүйелері. Бағдарламалаудың негізі

1.7 Соңғы реквизиттер

"Нақты уақыттың операциялық жүйелері" пәнін оқу кезіндегі алынған білім келесі пәндерді меңгеру кезінде қолданылады: ақпараттық технологиялар, автоматтандырылған жобалау жүйелері, басқару жүйесіндегі микропроцессорлық кешендер, технологиялық процестерді қолайлы басқару, SCADA - жүйелері

1.8 Пәннің мазмұны

1.8.1 Сабақтың түрлері және еңбек сыйымдылығы бойынша пәннің мазмұны

Бөлімнің аты	Сабақ түрлері бойынша еңбек сыйымдылығы, сағ				
	Дәріс-тер	тәжірбие-лік	зертхана-лық	СОДЖ	СДЖ
1 НУОЖ-нің басты түсініктері мен анықтамалары. НУОЖ мен жалпы қолданыстағы ОЖ-нің айырмашылықтары. НУОЖ құрылу ерекшеліктері. Нақты уақыт механизмдері					
Дәріс 1. НУОЖ-нің басты түсініктері мен анықтамалары. НУОЖ мен жалпы қолданыстағы ОЖ-нің айырмашылықтары	3	-	5	10	10
Дәріс 2. НУОЖ құрылу ерекшеліктері. Нақты уақыт механизмдері. НУОЖ-ның қолдану аясы					
2 UNIX ОЖ					
Дәріс 3. UNIX ОЖ. Жүйенің ерекшеліктеріне жалпы шолу					
Дәріс 4. UNIX ОЖ. Файлдық жүйе. Желілік өзара әсерлесулердің базалық механизмдері. Жадыны басқару.	6	-	5	10	10
Дәріс 5. UNIX ОЖ-дегі процестер мен тізбектерді, енгізу/шығаруды басқару					
3 Windows NT. «Классикалық» және объекті-бейімделген НУОЖ.					
Дәріс 6. Windows NT. Басты сипаттамалары. Windows NT ОЖ НУОЖ ретінде. Windows NT ОЖ үшін нақты уақыт кеңейтілулері	6	-	5	10	10

Бөлімнің аты	Сабақ түрлері бойынша еңбек сыйымдылығы, сағ				
	Дәріс-тер	тәжірбие лік	зертхана лық	СОДЖ	СДЖ
Дәріс 7. «Классикалық» және объекті-бейімделген НУОЖ.					
БАРЛЫҒЫ:	15	-	15	30	30

1.9 Негізгі әдебиет тізімі

1. А.В.Гордеев, А.Ю.Молчанов. Системное программное обеспечение. — "Питер", 2002. — 736с.
2. Мартин ДЖ. Программирование для вычислительных систем реального времени. Пер. с англ. Изд-во "Наука", 1975.- 360с.
3. Кристиан К. Введение в операционную систему Unix: пер. с англ. — М. Финансы и статистика, 1985. — 360с.
4. Робачевский А.М., Немнюгин С.А., Стесик О.Л. Операционная система Unix. 2-е изд.– СПб.: БХВ – Петербург, 2005. – 635с.
5. Соломейчук В.Г. Linux. Экспресс курс. – СПб.: БХВ – Петербург, 2005. – 277с.
6. Стахнов А.А. Linux. Научное издание. 2-е изд. - СПб.: БХВ – Петербург, 2005. – 944с.
7. Зыль С.Н. Операционная система реального времени QNX: от теории к практике. 2-е изд., перераб. доп. – СПб.: БХВ – Петербург, 2004. – 191с.
8. Тенанбаум Э. Современные операционные системы. пер. с англ. 2-е изд. – М.: СПб.: Нижний Новгород: Питер, 2005. – 1037с.
9. Олифер В.Г. Олифер Н.А. Сетевые операционные системы. М.: СПб.: Нижний Новгород: Питер, 2006. – 538с.
10. Грибанов В.П., Дробин С.В., Медведев В.Д. Операционные системы. - М.: Финансы и статистика, 1990. - 239 с.
11. Дейтел Х.М., Чофнес Р.Д. Операционные системы. пер. с англ. – М.: БИНОМ, 2006. – 704с.
12. Культин Н.Б. Delphi 5. Программирование на Object Pascal. — СПб.: ВHV — Санкт-Петербург, 1999.— 446 с., ил.
13. Культин Н.Б. Основы программирования в Delphi 7. — СПб.: БХВ — Санкт-Петербург, 2005.— 598 с., ил.
14. Климова Л.М. Delphi 7. Основы программирования. Решение типовых задач. 2-е изд. доп. – М.: Кудиц-Образ, 2005. – 480с.
15. Дарахвелидзе П.Г., Марков Е.П. Программирование в Delphi 7. - СПб.: БХВ — Петербург, 2005.— 781 с.
16. Бобровский С.И. Delphi 7. – М.: СПб.: Нижний Новгород: Питер, 2005. – 735с.
17. Кузменко В.Г. Visual Basic 6: самоучитель. 3-е изд. - М.: ООО «Бином-Пресс», 2005. – 400с.
18. Михаэль Райтингер, Геральд Муч. Visual Basic 6: полное руководство пер. с нем. — К.: Издательская группа ВHV, 2000. — 720 с.
19. Роман С. Visual Basic. Библиотека WIN 32 API. пер. с англ. – М.: ДМК Пресс, 2005. – 476с.
20. Курилович В. Visual Basic. – М.: СОЛОН-Пресс, 2005. – 383с.
21. Пахомов Б.И. C/C++ и Borland C++ Builder для начинающих. - СПб.: БХВ — Петербург, 2006.— 628с.

22. Культин Н. С++ Builder в задачах и примерах. - СПб.: БХВ —Петербург, 2005.— 327с.

23. Лифты. Учебник для вузов/ Под общей ред. Д.П. Волкова – М.: изд-во АСВ 1999.-480с. с ил.

24. Полковников В.С. и др. Монтаж лифтов. Учебник для сред. Проф.-тех. училищ. Изд. 3-е, перераб. и доп. – М.: Высшая школа 1986. –304с. с ил.

1.10 Қосымша әдебиеттер тізімі

1. Партыка Т.Л., Попов И.И. Операционные системы, среды и оболочки. М.: ФОРУМ: ИНФРА-М, 2005. – 399с.

2. Соловьев Г.Н., Никитин В.Д. Операционные системы ЭВМ. - М.: Высшая школа, 1989.-255 с.

3. Дейтел Г. Введение в операционные системы. В 2-х т. Т.1. - Пер с англ. - М: Мир, 1987. - 359 с.

4. Операционная система реального времени QNX Neutrino 6.3. Системная архитектура. пер. с англ./ гл. ред. Е. Кондукова. – М.: СПб.: Нижний Новгород: Питер, 2006. – 336с.

5. Зыль С.Н. QNX momentics. Основы применения. – СПб.: БХВ – Петербург, 2005. – 255с.

6. Болл Б., Хойт Д. Red Hat Linux.8/9. Настольная книга пользователя. пер. с англ. – М.: СПб.: Киев: DiaSoft, 2005. – 921с.

7. Цилюрик О., Горшко Е. QNX/UNIX. Анатомия параллелизма. – СПб.: М.: Символ, 2006. – 287с.

1.11 Студенттердің білімін бағалау өлшемі

Пән бойынша емтихандық баға шекаралық бақылаудың максималдық үлгерім көрсеткіштерінің сомасы (60% дейін) және қорытынды аттестация (емтиханнан) (40% дейін) түрінде анықталады және кестеге сай 100% дейін маңыздылықты құрайды.

Әріптік жүйемен бағалау	Балдар	%-дық мазмұны	Дәстүрлі жүйемен бағалау
A	4,0	95-100	Өте жақсы
A-	3,67	90-94	
B+	3,33	85-89	Жақсы
B	3,0	80-84	
B-	2,67	75-89	
C+	2,33	70-74	Қанағаттанарлық
C	2,0	65-69	
C-	1,67	60-64	
D+	1,33	55-59	
D	1,0	50-54	Қанағаттанғысыз
F	0	0-49	

Шекаралық бақылау 5-ші,10-шы,15-ші оқу апталарында өткізіледі және келесі бақылау түрлерінен қалыптасады:

Бақылау түрі	% - дық мазмұны	Академиялық оқу мерзімі, апта															Барлығы, %
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Сабаққа қатысу	15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	15
Лаб. жұмыстар	20		3		3		3		3			3				5	20
СОДЖ	10			2			2			2			2		2		10
Модуль	15					5					5					5	15
Емтихан	40																40
Барлығы	100	1	4	3	4	6	6	1	4	3	6	5	2	1	3	11	60

1.12 Саясат пен процедуралар

"Нақты уақыттың операциялық жүйелері" пәнін оқығанда келесі ережелерді орындауды сұраймын:

1. Сабаққа кешікпеу .
2. Орынды себептерсіз сабақты жібермеу, ауырған жағдайда анықтама көрсету, басқа жағдайларда түсіндірме хатты ұсыну.
3. Дайын болу: дәріс сабақтарында өткен дәріс материалдарынан, СОДЖ-да қойылған сұрақтардан, лабораторлық сабақтарда – мақсатарды және ағымды жұмыстың жүрісін білу .
4. Дәріс конспекттерін ұқыпты жүргізу, КарМТУдың ережелері мен талаптарына сәйкес лабораторлық жұмыстарды жасау .
5. Оқу барысына активті қатысу.
6. Курстастарға және оқытушыларға шыдамды, ашық, шын көңілді және тілектес болу.

1.13 Пәннің оқу-әдістемелік қамтамасыз етілгені

Автордың аты-жөні	Оқу-әдістемелік әдебиеттің атауы	Баспа, басылым жылы	Дана саны	
			кітапхан адағы	кафедрадағы
Негізгі әдебиеттер:				
А.В.Гордеев, А.Ю.Молчанов	Системное программное обеспечение.	"Питер", 2002. — 736с.	2	-
Мартин ДЖ.	Программирование для вычислительных систем реального времени.	Пер. с англ. Изд-во "Наука", 1975.- 360с.	1	-
Кристиан К.	Введение в операционную систему Unix. пер. с англ.	М. Финансы и статистика, 1985. – 360с.	13	-
Робачевский А.М., Немнюгин С.А., Стесик О.Л.	Операционная система Unix. 2-е изд.	СПб.: БХВ – Петербург, 2005. – 635с.	5	-

Автордың аты-жөні	Оқу-әдістемелік әдебиеттің атауы	Баспа, басылым жылы	Дана саны	
			кітапхан адағы	кафедрадағы
Соломейчук В.Г.	Linux. Экспресс курс.	СПб.: БХВ – Петербург, 2005. – 277с.	1	-
Стахнов А.А.	Linux. Научное издание. 2-е изд.	СПб.: БХВ – Петербург, 2005. – 944с.	4	-
Зыль С.Н.	Система реального времени QNX: от теории к практике. 2-е изд., перераб. доп.	СПб.: БХВ – Петербург, 2004. – 191с.	6	-
Тенанбаум Э.	Современные операционные системы. пер. с англ. 2-е изд.	М.: СПб.: Нижний Новгород: Питер, 2005. – 1037с.	1	-
Олифер В.Г. Олифер Н.А.	Сетевые операционные системы.	М.: СПб.: Нижний Новгород: Питер, 2006. – 538с.	1	-
Грибанов В.П., Дробин С.В., Медведев В.Д.	Операционные системы.	М.: Финансы и статистика, 1990. - 239 с.	15	-
Дейтел Х.М., Чофнес Р.Д.	Операционные системы. пер. с англ.	М.: БИНОМ, 2006. – 704с.	15	-
Культин Н.Б.	Delphi 5. Программирование на Object Pascal.	СПб.: BHV — Санкт-Петербург, 1999.— 446 с., ил.	6	-
Культин Н.Б.	Основы программирования в Delphi 7.	СПб.: БХВ — Санкт-Петербург, 2005.— 598 с., ил.	6	-
Климова Л.М.	Delphi 7. Основы программирования. Решение типовых задач. 2-е изд. доп.	М.: Кудиц-Образ, 2005. – 480с.	3	-
Дарахвелидзе П.Г., Марков Е.П.	Программирование в Delphi 7.	СПб.: БХВ — Петербург, 2005.— 781 с.	2	-
Бобровский С.И.	Delphi 7.	М.: СПб.: Нижний Новгород: Питер, 2005. – 735с.	3	-
Кузменко В.Г.	Visual Basic 6: самоучитель. 3-е изд.	- М.: ООО «Бином-Пресс», 2005. – 400с.	3	-
Михаэль Райтингер, Геральд Муч.	Visual Basic 6: полное руководство пер. с нем.	К.: Издательская группа BHV, 2000. — 720 с.	3	-
Роман С.	Visual Basic. Библиотека WIN 32 API. пер. с англ.	М.: ДМК Пресс, 2005. – 476с.	5	-
Курилович В.	Visual Basic.	М.: СОЛОН-Пресс, 2005. – 383с.	5	-
Пахомов Б.И.	C/C++ и Borland C++ Builder для начинающих.	СПб.: БХВ — Петербург, 2006.— 628с.	2	-

Автордың аты-жөні	Оқу-әдістемелік әдебиеттің атауы	Баспа, басылым жылы	Дана саны	
			кітапхан адағы	кафедрадағы
Культин Н.	C++ Builder в задачах и примерах.	СПб.: БХВ — Петербург, 2005.— 327с.	2	-
Қосымша әдебиеттер				
Партыка Т.Л., Попов И.И.	Операционные системы, среды и оболочки.	М.: ФОРУМ: ИНФРА-М, 2005. – 399с.	1	-
Соловьев Г.Н., Никитин В.Д.	Операционные системы ЭВМ.	М.: Высшая школа, 1989.-255 с.	25	-
Дейтел Г.	Введение в операционные системы. В 2-х т. Т.1.	Пер с англ. - М: Мир, 1987. - 359 с.	1	-
гл. ред. Е. Кондукова.	Операционная система реального времени QNX Neutrino 6.3. Системная архитектура. пер. с англ.	М.: СПб.: Нижний Новгород: Питер, 2006. – 336с.	1	-
Зыль С.Н.	QNX momentics. Основы применения.	СПб.: БХВ – Петербург, 2005. – 255с.	4	-
Болл Б., Хойт Д.	Red Hat Linux.8/9. Настольная книга пользователя. пер. с англ.	М.: СПб.: Киев: DiaSoft, 2005. – 921с.	2	-
Цилюрик О., Горшко Е.	QNX/UNIX. Анатомия параллелизма.	СПб.: М.: Символ, 2006. – 287с.	1	-

2. Пән бойынша тапсырмаларды орындау және тапсыру кестесі

Бақылау түрі	Тапсырманың мазмұны мен міндеті	Ұсынылатын әдебиет	Орындалу уақыты, сағ	Бақылау формасы	Тапсыру уақыты
Зертханалық жұмыс № 1	Көтерме орнатудың программалық имитаторын жасау. Көтерме орнату құрылғылары жұмысымен танысу. Компонент жұмысы мен оның қасиеттерін зерттеу.	12-24	2	Лабораториялық жұмыс бойынша есеп	Оқудың 2-ші аптасы
Зертханалық жұмыс № 1	Көтерме орнатудың программалық имитаторын жасау. Көтерме орнату құрылғылары жұмысымен танысу. Берілген нұсқа бойынша көтерме орнатудың программалық жұмысын құру.	12-24	2	Лабораториялық жұмыс бойынша есеп	Оқудың 4-ші аптасы

Бақылау түрі	Тапсырманың мазмұны мен міндеті	Ұсынылатын әдебиет	Орындалу уақыты, сағ	Бақылау формасы	Тапсыру уақыты
Зертханалық жұмыс № 2	Технологиялық процестің имитаторын жасау. Берілген нұсқа бойынша технологиялық процестің жұмысын зерттеу.	12-22	2	Лабораториялық жұмыс бойынша есеп	Оқудың 6-ші аптасы
Лабораторная работа № 2	Технологиялық процестің имитаторын жасау. Технологиялық процестің виртуальды имитаторының графикалық формасын жұмысын жасау.	12-22	2	Лабораториялық жұмыс бойынша есеп	Оқудың 8-ші аптасы
Лабораторная работа № 2	Технологиялық процестің имитаторын жасау. Технологиялық процестің имитаторы үшін басқару панелін, апатты және бақылау панелін жасау.	12-22	3	Лабораториялық жұмыс бойынша есеп	Оқудың 11-ші аптасы
Лабораторная работа № 2	Технологиялық процестің имитаторын жасау. Технологиялық процестің виртуальды жадысының жұмысын программалау.	12-22	4	Лабораториялық жұмыс бойынша есеп	Оқудың 15-ші аптасы

3 Дәрістер конспекті

1 бөлім. НУОЖ негізгі түсініктері мен анықтамалары. НУОЖ мен жалпы қолданыстағы ОЖ айырмашылықтары. НУОЖ құрылу ерекшеліктері. Нақты уақыт механизмдері.

1 Тақырып. НУОЖ негізгі түсініктері мен анықтамалары. НУОЖ мен жалпы қолданыстағы ОЖ айырмашылықтары.

Дәріс жоспары

НЕГІЗГІ АНЫҚТАМАЛАР

Нақты өлшемдегі уақытты қолдану.

Нақты уақыт жүйесінің анықтамалары

НУОЖ ТОПТАЛУЫ (КЛАССИФИКАЦИЯСЫ)

Реакция уақыты бойынша

Бағдарламалық жабдықтаудың зерттемесінің тәсілі бойынша

Шығу тегіне байланысты

Жүйелік бағдарламалық орта бойынша

Ішкі құрылысы бойынша

НУОЖ-ГЕ ҚОЙЫЛАТЫН ТАЛАПТАР МЕН ЕРЕКШЕЛІКТЕР

НУОЖ МЕН ЖАЛПЫ ҚОЛДАНЫСТАҒЫ ОЖ АЙЫРМАШЫЛЫҚТАРЫ.

1.1 Негізгі анықтамалар

Қазіргі кезде «Өзіндік байланысы бар жүйелер» (on line) және «нақты уақытта жұмыс істейтін жүйелер» (real-time) деп аталатын жүйелер кеңінен таралған.

Көптеген түрлі ЭЕМ-ге жіберілетін мәліметтер мен өңделген мәліметті қабылдайтын құрылғылар *терминалдар* деп аталады.

Өзіндік байланысы бар жүйе келген мәліметтер ЭЕМ-ға оның пайда болу нүктесінен тікелей енгізілетін және немесе келген мәліметтер олар қолданылатын жерге тікелей жіберілетін жүйе ретінде анықталуы мүмкін.

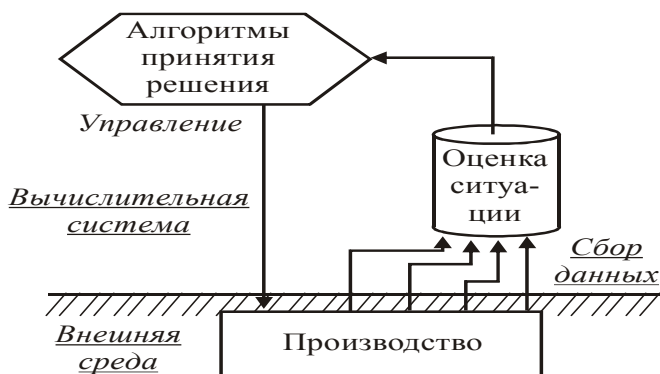
Нақты уақытта жұмыс істейтін есептеуіш жүйе сыртқы объектілерді басқарып, сол уақыт мезетінде сыртқы объектілердің қызмет етуіне әсер ету үшін мәліметтерді қабылдап, оларды өңдеп және жеткілікті нәтижені тез қайтаратын жүйе ретінде анықталуы мүмкін.

1.1.1 Нақты өлімдегі уақытты қолдану

Нақты уақыт жүйесінің қолдану аясын 2 топқа бөлуге болады:

1. Бірінші топқа нақты уақытта жұмыс істейтін өзіндік байланыс жүйесі бар бұрын қолдан немесе машина көмегімен істелінетін және тиімдірек және үнемді орындалатын жұмыстар жатады.
2. Екінші қызықтырақ топқа бұрын орындалуы мүмкін емес болған жұмыстар үшін жаңа жүйелердің қолданылуы жатады.

Өндірістік кәсіпорындарда көптеген есептер мен есептеулерге арналған сырттан келген мәліметтер кәсіпорындар цехтарынан келіп түседі. Осы барлық мәліметті сақтайтын нақты уақыт жүйесі басшылыққа қажетті шешімді тез және қатесіз қабылдауға мүмкіндік береді.



1 сурет. Нақты уақыттағы кәсіпорынды басқару жүйесі

Бұлай жұмыс істей отырып, ЭЕМ басқару функцияларын орындайды.

Нақты уақыт жүйесінің мақсаты кейбір процес тиімді өту үшін әр түрлі шешімдер мен әрекеттерді біріктіру болып табылады.

Жүйе жетілдірілуі үшін:

- кәсіпорынның жоспарлану циклін азайту
- айналымды барынша күшейту
- Ұйымның мүмкіндіктерін тиімді қолдануға қол жеткізу
- Мәліметтерді қажет ететін жерлерге уақытында жеткізу

Жүйені жобалау мен жоспарлау кезінде жүйе реакциясының әр түрлі уақытта қамтамасыздандырылуының бағасын ескеру қажет.

1.1.2 Нақты уақыт жүйесінің анықтамалары

Нақты уақыт жүйесіне қысқаша анықтама беріп көрейік:

1. **Жүйе нақты уақыттағы жүйе (НУЖ) деп аталады**, егер оның қызметінің дұрыстығы есептеулердің логикалық мүлтіксіздігіне ғана емес, сонымен қатар осы есептеулер шығарылатын уақытқа тәуелді болса.

2. **Жүйе нақты уақыттағы жүйеде жұмыс істейді делінеді**, егер оның тез әсер етуі бақылау мен басқару объектілеріндегі физикалық процестердің өту жылдамдығымен пара-пар болса. Бұл жерде нақты уақыт жүйесімен орындалатын функциялармен тікелей байланысты процестер туралы айтылады. Басқару жүйесі мәліметтерді жинау, берілген алгоритмге сәйкес өңдеу жүргізу және жүйеге қойылатын міндеттің сәтті орындалуын қамтамасыз ететін уақыт аралығында басқарушы әсер беруі керек.

1.2 НУОЖ топталуы (классификациясы)

1.2.1 Реакция уақыты бойынша

Реакция уақыты бойынша бөлінеді:

- **Қатты (hard)** нақты уақыт жүйелері
- **Жұмсақ (soft)** нақты уақыт жүйелері

Қатты нақты уақыт жүйелері қандай жағдай болса да жүйе реакциясының ешқандай кідіріс жасауын болдырмайды, өйткені:

1. Кешігу жағдайында нәтиже тиімсіз болады
2. Реакцияның кідіріс жасау жағдайынан апат болуы мүмкін
3. Кешігу күні шексіз үлкен болуы мүмкін

Қатты нақты уақыт жүйелері мен жұмсақ нақты уақыт жүйелері арасындағы негізгі айырмашылықты былай суреттеуге болады: қатты нақты уақыт жүйелері реакциямен бірге оқиғаға ешқашан кешікпейді, ал жұмсақ нақты уақыт жүйелері реакциямен бірге оқиғаға кешікпеуі тиіс.

1.2.2 Бағдарламалық жабдықтаудың зерттемесінің тәсілі бойынша

Бағдарламалық жабдықтаудың зерттемесінің тәсілі бойынша оларды келесі топтарға бөледі:

1. **Self-Hosted** НУОЖ – бұл НУОЖ өзінде жұмыс істей отырып, тұтынушылар қосымшаларды ойлап таба алатын жүйелер.

Мұндай жүйелердің *жетістігі*- тұтынушылар жұмыс істейтін машинаны қолданатын қосымшаның құру және іске қосудың қарапайым және көрнекі механизмі болып саналады.

Кемшілігі өнеркәсіптік компьютер нақты пайдалану кезінде тұтынушы интерфейсі мен компилятор сияқты ауырсалмақты бағдарламаларды іске қосу мүмкіндігін қажет етпейді.

2. **Host/Target** НУОЖ –бұл қосымшалар жетілдірілетін (host) ОЖ мен компьютер және қосымшалар іске қосылатын (target) ОЖ мен компьютерлерден тұратын жүйелер, олар әр түрлі болып келеді.

Осы түрдегі жүйелер бірқатар жетістіктерге ие: олардың ішіндегі ең бастысы – жылдамдық пен жүйенің реактивтілігі. Осы түрдегі жүйенің жоғарғы реактивтілігінің басты себебі- ағынның болуы, сондықтан араларында контекстің ауысуы өте аз уақыт кетеді (процеске қарағанда). Сонымен қатар мұндай жүйелердің жетістігі қосымшаға керекті компоненттерді қосу арқылы НУОЖ өлшемдерін кішірейту және қосымшалар құру үшін «қарапайым» жүйенің барлық қорын пайдалану (графикалық интерфейс, бумалық жүйе, жылдам процессор, оперативті жадының үлкен көлемі) болып табылады.

Осы басты жетістіктермен бірқатар *кемшіліктер* байланысты: желі тоқтап қалғанда бүкіл жүйенің тоқтауы, жаңа қосымшалардың динамикалық ПОДГРУЗКА мәселелері.

Бағдарламалық компоненттің салыстырмалы қиындығы кемшілік болып табылады: кросс-компиляторлар, жойылған тиеушілер және жөндеулер және т.б.

1.2.3 Шығу тегіне байланысты

Шығу тегіне байланысты НУОЖ келесі топтарға бөлінеді:

1. НУОЖ ретінде қолданылатын **кәдімгі ОЖ**
2. **Меншікті** НУОЖ. Нақты уақыттағы есептерді қолдануға арналған мамандандырылған операциялық жүйелер. Олар self-hosted, host/target түрінде болады, кейбір НУОЖ 2 модельді де қолданады.
3. **Мамандандырылған** (жеке меншік) НУОЖ. Бұл өндірушінің нақты микроконтролері үшін жетілдірілген НУОЖ. ОЖ көптеген жағдайда сапалы болмайды, ол қосымшасы бар бірлік модульді береді және қажетті минимум қызметті ғана қамтамасыздандырады. Мұндай жүйелер host/target сатысына жатады.

1.2.4 Жүйелік бағдарламалық орта бойынша

Нақты уақыт міндеттерін ерекше жүйелік бағдарламалық орта шеңберінде іске асыру керек екені анық болып отыр. Нақты уақыт жүйелерін 4 класқа бөлуге болады:

1-нші класс: микропроцессорлар деңгейінде бағдарламалау

2-нші класс: Нақты уақыт жүйесінің ең аз ЯДРОСЫ???

3-нші класс: Нақты уақыт жүйесінің ядросы және аспапты орта

4-нші класс: Толық сервисті ОЖ

2-нші және 3-нші кластағы жүйелерді «қатты» деп, ал 4 классты – «жұмсақ» нақты уақыт жүйелері деп атайды.

1.2.5 Ішкі құрылысы бойынша

Бөлінеді:

1. Классикалық
2. Объекті-бағытталған жүйелер. Олардың қатарына объекті-бағытталған тілге арналған зерттеме құралдары мен кітапхана қоры ғана емес, осы тілде өздерінің жазылған жүйелері жатады.

1.3 НУОЖ-ге қойылатын талаптар мен ерекшеліктер

НУОЖ-ге қойылатын басты талаптар төмендегідей:

- уақыт бойынша талаптар
 - бірнеше міндеттің қатар (паралельді) орындалу мүмкіндігі
 - алдын-ала болжау мүмкіндігі
 - оқиғаға үндеудің орташа ғана емес, барынша маңыздылығы
 - қауіпсіздік мәселелеріндегі басты талаптар
 - ұзақ уақыт мезетінде сапалы жұмыс мүмкіндігі
- НУОЖ жалпы сипаттамасы:
- үлкен және күрделі жүйелер
 - реттелген жүйелер
 - аппаратурамен қатал байланыс
 - міндеттің орындалуы уақытқа тәуелді
 - тестілеудегі күрделілік

1.4. НУОЖ мен жалпы қолданыстағы ОЖ айырмашылықтары.

1. Жалпы қолданыстағы ОЖ, әсіресе UNIX сияқты көп қолданылатын, тұтынушы мен орындалатын процестер арасындағы компьютердің қолайлы реттелген қорларына бағытталған. НУОЖ-де осындай міндеттер екінші қатарға қалады, барлығы басты мақсатқа шегінеді-объекте болатын оқиғаларға төтеп беріп үлгеру.

2. Келесі айырмашылығы НУОЖ қолданылуы объекте болатын оқиғалар мен мен аспаптармен әрқашан байланысты болуы. Нақты уақыт жүйесі аппаратты-бағдарламалық кешен сияқты төмендегілерден тұрады:

- Көрсеткіштер, объектілерде тіркелетін оқиғалар
- компьютерде өңдеуге жарамды көрсеткіштерді сандық түрге келтірілетін енгізу-шығару модулі

- объектідегі оқиғаларға көңіл бөлетін бағдарламасы бар компьютер

НУОЖ ішкі оқиғаларды өңдеуге негізделген. Атап айтқанда, бұл жалпы қолданыстағы ОЖ негізгі айырмашылығына себепті болатын:

- құрылымы бойынша
- ядроның қызметі бойынша
- ендіру-шығару жүйесінің құрылу бойынша

3. Бұдан басқа, НУОЖ қолданылуы әрқашан нақты. Егер жалпы қолданыстағы ОЖ дайын қосымша жиынтығы тұтынушылармен қолданылса, онда НУОЖ нақты уақыттағы аппараты- бағдарламалық кешенді құру үшін тек құрал ретінде қызмет етеді.

Белгісіз ішкі оқиғалар ағынына белгілі уақыт кезеңінде әсер ететін аппаратты – программалық кешенді нақты уақыт жүйесі деп атаймыз.

Бұл ***анықтама*** келесіні түсіндіреді:

1. Біріншіден, жүйе осы оқиғаға дағдарыс уақыт аралығында осы объектіде болған оқиғаларға қарсы тұрып үлгеру керек. (meet deadline)
2. Екіншіден, жүйе бір мезгілде болатын оқиғаларға көңіл бөліп үлгеру керек.

СДЖ арналған бақылау жұмыстары:

- 1 Терминалдар дегеніміз не? Мысал келтір
- 2 Нақты уақытта жұмыс атқаратын жүйелер өзіндік байланысы бар жүйелер арасында қандай айырмашылық бар?
- 3 Нақты уақыт жүйесін қолданудың қандай 2 тобы (категориясы) бар?
- 4 Нақты уақыт жүйесінің мақсаты не болып табылады?
- 5 Нақты уақыт жүйесіне бірнеше анықтама бер. Олардың қайсысы нақтырақ? Не себепті?
- 6 Жұмсақ нақты уақыт жүйесі деген не?
- 7 Қатты нақты уақыт жүйесі деген не?
- 8 Қатты және жұмсақ нақты уақыт жүйелері арасында қандай айырмашылықтар мен ұқсастықтар бар?
- 9 Self-Hosted НУОЖ. Оның артықшылықтары мен кемшіліктері.
- 10 Host/Target НУОЖ. Оның артықшылықтары мен кемшіліктері.
- 11 Шығу тегіне байланысты НУОЖ қандай түрлері болады? Олар қандай түрге жатады және неліктен?
- 12 Жүйелік бағдарламалық ортаға байланысты НУОЖ қандай кластарға бөлінеді?
- 13 Ішкі құрылымы бойынша НУОЖ қандай болады?
- 14 НУОЖ мен жалпы қолданыстағы ОЖ негізгі айырмашылықтары

Ұсынылатын әдебиеттер:

1. Мартин ДЖ. Программирование для вычислительных систем реального времени. Пер. с англ. Изд-во "Наука", 1975.- 360с.
2. Тенанбаум Э. Современные операционные системы. пер. с англ. 2-е изд. – М.: СПб.: Нижний Новгород: Питер, 2005. – 1037с.
3. Олифер В.Г. Олифер Н.А. Сетевые операционные системы. М.: СПб.: Нижний Новгород: Питер, 2006. – 538с.
4. Грибанов В.П., Дробин С.В., Медведев В.Д. Операционные системы. - М.: Финансы и статистика, 1990. - 239 с.

5. Дейтел Х.М., Чофнес Р.Д. Операционные системы. пер. с англ. – М.: БИНОМ, 2006. – 704с.

15 тақырып. НУОЖ құрылу ерекшеліктері. Нақты уақыт механизмі. НУОЖ қолдану аясы

Дәріс жоспары

- 15.1 НУОЖ құрылу ерекшеліктері
 - 15.1.1 НУОЖ-дегі орындау және жетілдіру жүйесі
 - 15.1.2 Ядро
 - 15.1.2.1 Қордың синхрондылығы
 - 15.1.2.2 Міндетаралық айырбас
 - 15.1.2.3 Мәліметтердің бөлінуі
 - 15.1.2.4 Сыртқы құрылғылардың сұраныстарының өңделуі
 - 15.1.2.5 Ерекше жағдайлардың өңделуі
 - 15.1.3 ОЖ жоспарлаудың әдістері
 - 15.2 Нақты уақыт механизмі
 - 15.2.1 Басымдық жүйесі және диспетчерлеу алгоритмі
 - 15.2.2 Міндетаралық өзара әсер механизмі
 - 15.2.3 Жүйенің реакция уақыты
 - 15.2.4 Нақты уақыт ядролары
 - 15.2.5 Жүйе өлшемдері
 - 15.2.5.1 ПЕСК-дағы жүйенің орындалу мүмкіндігі
 - 15.2.6 Контекстінің ауыстыру уақыты
 - 15.2.7 Таймермен жұмыс істеу құрылғылары
- 2.4 НУОЖ қолдану аясы

НУОЖ құрылу ерекшеліктері

2.1.1 НУОЖ-дегі орындау және жетілдіру жүйесі

НУОЖ-ң жалпы қолданыстағы жүйеден негізгі айырмашылығының бірі жетілдіру жүйесі мен орындау жүйесінің нақты шектелуі.

НУОЖ орындау жүйесі- нақты уақыт қосымшаларының қызметін қамтамасыздандыратын құралдар жиынтығы (ядро, драйверлер, орындаушы модульдер).

НУОЖ –н орындалу ортасына қойылатын талаптар:

- Жүйенің шағын жадысы- оны енгізу мүмкіндігі үшін;
- Жүйе жадыда толығымен резидентті болуы керек;
- Жүйенің барлық ресурстарын барынша тиімді орындауды қамтамасыз ету үшін жүйе ***көпміндетті*** болуы керек;
- Үзіліссіздіктерді қамтамасыздандыруға арналған басымды ядролар
- Басымды диспетчерлер-жүйеге қатысты емес артықшылықты қолданбалы бағдарламалар жетілдірушіге әр енгізу модулін иеленуге мүмкіндік береді.

Жетілдіру жүйесі- нақты уақыт қосымшасының құрылуын қамтамасыз ететін құрылғылар жиынтығы.

НУОЖ жетілдіру құралдары атқаратын қызметі бойынша Developers Studio, TaskBuilder сияқты үйреншікті жетілдіру жүйелерінен айырмашылықтары бар екенін ескеруге болады, себебі оның құрамында:

- Жойылған жөндеу құрылғылары
- Кескінделген құралдар (кодтың бөлінген бөлігінің орындалу уақытын өлшеу)
- Тұтас процессордың эмуляция құралдары
- Жөндеудің өзара әсерлесуші міндеттерінің арнайы құралдары

- ал кейбір кезде модельдеу құралдары бар

Ядро

Ядро тұтас компьютердің аппараттық құрылғыларын басқарады:

- орталық процессорды
- енгізу/ шығару құралдары мен жады
- қолданбалы сипатты бағдарламалық құралдардың және басқа жүйелердің жұмысын бақылайды.

Ядро (Kernel) мен ОЖ арасында нақты шекара жоқ. Оларды ережеге сәйкес, атқаратын қызметінің мүмкіндіктері бойынша ажыратады.

Ядро тұтынушыға төмендегідей базалық құрылғыларды ұсынады:

- міндеттердің синхрондылығын жоспарлау
- міндетаралық коммуникация
- жадыны басқару және т.б

Сонымен қатар операциялық жүйелер мынадан тұрады:

- бумалық жүйе
- желілік тірек
- операторлы интерфейс
- жоғары дәрежелі басқа құрылғылар

Ядро 5 түрдегі қызмет көрсетуді қамтамасыз етеді.

Қордың синхрондылығы

Синхрондау әдісі жалпы қорға қол жеткізуді шектеуді талап етеді (мәліметтер мен сыртқы құрылғыларға). Қарапайым синхрондаудың кеңінен таралған түрі – жалпы қорларға таңдаулы қол жеткізуді қамтамасыз ететін екілік семафор. Үлкен қатеге тұрақты қабілеті бар жүйелер есептік семафорды иеленуі мүмкін. Семафордың бұл түрі тек процесстің анықталған мөлшеріне ғана қорға бір мезгілде қол жеткізуге рұқсат етеді.

Міндетаралық айырбас

Сыртқы байланыс хатты тасымалдау жүйесі арқылы жүзеге асырылуы мүмкін. Ішкі байланысты бірдей жүйелер ішінде датаграмма арқылы, байланыс сызықтары арқылы ұйымдастыруға болады. Бірдей әдісті таңдау хаттама байланысына тәуелді.

2.1.2.3 Мәліметтердің бөлінуі

Көптеген жүйелерде жадының жалпы бөліміне қол жеткізу көзделген. Мәліметтердің кезекті ұйымы кеңінен таралған. Кезектің көп түрлері қолданылады, олардың әрқайсысының өзіндік артықшылықтары бар.

Сыртқы құрылғылардың сұраныстарының өңделуі

Ядро қолданбалы бағдарламаларға құрылғылардан оқу оларға жазуды іске асыратын енгізу/шығару қызметтерін қамтамасыз ету керек.

Ерекше жағдайлардың өңделуі

Ерекше жағдай бағдарламаны орындау кезінде туындайтын оқиғаны береді. Ерекше жағдай өндеудің 2 әдісі бар – қате шарттарды анықтауға арналған күйлердің маңызын қолдану және ерекше жағдай өндеушіні қате шарттарды үзу және оларды түзеу үшін қолдану.

ОЖ жоспарлаудың әдістері

Кез келген НУОЖ маңызды бөлігі міндеттерді жоспарлаушы болып табылады, оның қызметі – нақты уақыт мезетінде жүйеде қандай қызмет орындалу керек екенін анықтау.

Негізінен басты **жоспарлау тәсіліне:**

- циклдік алгоритм (round robin стилінде)
- тең қол жеткізімді уақыттың жіктелінуі (time sharing with fairness)
- кооперативті көпміндеттілік

НУОЖ кеңінен қолданылатын жоспарлау принципі- ығыстырушы басымды көпміндеттілік. Негізгі идеясы- жоғарғы артықшылықты міндет жұмыс табылса, төменгі басымды жұмысты жоққа шығарады.

Жалпы алғанда, жоспарлау алгоритмдері жүйе қызметінің қолайлы шарттарына сай келуі керек. Бірақ егер қатты нақты уақыт жүйелеріне « әрқашан және барын уақытында істеу» деген шарт анық болса, жұмсақ нақты уақыт жүйесі үшін мысалы ең аз барынша кешігу және операцияның аяқталуының ортаөлшемді мезгілдік.

Әр міндет жүйеде белгілі бір жеке қызмет атқарғанымен, көптеген жағдайда әр түрлі міндеттерді орындайтын әрекеттердің үйлесімділігінде қажеттілік туады. Бұндай синхрондылық негізінен келесі жағдайларда қажет:

1. Түрлі міндет атқаратын функциялар бір-бірімен байланысты. Мысалы егер бір міндет екіншісіне шығыс мәліметтерін дайындап берсе, онда соңғысы бір міндеттің сәйкес хабар алмағанша орындалмайды. Бұл жағдайда вариациялардың бірі- міндеттің белгіленген шартта бір немесе бірнеше жаңа міндеттерді тудыруы.

2. Бірнеше міндеттің бөлінген қорға қол жеткізуді реттеу қажет.

3. Сыртқы оқиғалары бар міндеттерді синхрондау қажет. Ереже бойынша бұл үшін үзіліссіздік механизмі қолданылады.

4. Міндеттерді уақыт бойынша синхрондау қажет. Бұл жағдайда түрлі нұсқалардың диапазоны нақты астрономиялық уақыттағы белгілі бір әсердің белгіленген уақыт интервалы міндеттерінің орындалуының кідірілуіне дейін кең таралған. Ақырында бұл мәселелерді шешу үшін *таймер* деп аталатын арнайы аппаратты құралдар қолданылады.

Нақты уақыт механизмі

НУОЖ бағалаудағы маңызды параметр жүйе ұсынатын нақты уақыт механизмінің құралдар жиынтығы.

Басымдық жүйесі және диспетчерлеу алгоритмі

Жүйенің көріністі жетілдірудің базалық құралдары:

1. ***Процестің басымдық жүйесі***

2. НУОЖ ***жоспарлау алгоритмінен*** тұрады.

Көпміндетті жалпы қолданыстағы ОЖ-де үзіліссіз уақыт кванты түсінігіне негізделген, берілген процесті орындауға арнап ұсынылатын айналмалы диспетчерлеу алгоритмінің түрлі жаңартулар қолданылады. Жоспарлаушы әр уақыт квантының аяқталуы бойынша белсенді процестер кезегін қарастырады және басымдықтарын негізге ала отырып, олардың қайсысына басқаруды беретінін шешеді.

Басымдық

-бекітілген

-немесе уақыт бойынша ауысу мүмкін.

Жоспарлау алгоритмінің мүмкін түрі- ***ығыстырушы басымдық***

Міндетаралық өзара әсер механизмі

Нақты уақыт механизмінің басқа жиынтығы процестер синхрондылығы мен олардың арасындағы мәліметтер тасымалы құрылғыларына жатады. Оларға семафорлар, мьютекстер (mutex), оқиғалар, белгілер, бөлшекті жадымен жұмыс атқаратын құралдар, мәліметтер каналы (pipes), хабарлар кезегі.

2.2.3. Жүйенің реакция уақыты

Нақты уақыт жүйесін өндірушілердің көбісі жүйенің реакциясының үзіліссіздікке уақыты деген параметр келтіреді. Жүйе реакциясының уақытын болжау үшін біз қандай заманды білуіміз керек екенін түсінеміз.

- Көрсеткіштермен тіркелетін объектідегі оқиғалар. Мәліметтер көрсеткіштен енгізу/шығару модуліне жіберілетін жүйелер (интерфейстер)

- Енгізу/шығару модулі, көрсеткіштен мәліметті қабылдап, оны өндегеннен кейін, объектіде оқиға болған туралы белгі бере отырып, басқарма компьютерде сұранысты үзіліссіздікке зерттейді.
- Енгізу/шығару модулінен белгі алысымен, жүйе осы оқиғаны өңдейтін бағдарламаны іске қосу қажет.

Уақыт интервалы- объектідегі оқиғадан осы оқиғаны өңдейтін бағдарламадағы бірінші нұсқаулар орындалғанға дейін жүйе реакциясының уақыты болып табылады. Және нақты уақыт жүйелерін жобалау арқылы жетілдірушілер осы интервалды анықтай білу керек.

2.2.4 Нақты уақыт ядролары

Бұл класқа осы операциялық жүйелердің нақты уақыт механизмдерінің барлығының іске асырылатын монолитті ядросы бар жүйелер кіреді. Тарихи жүйенің бұл түрі өте жақсы жобаланған болатын. Ережеге сәйкес, нақты уақыт ядроларының жетілдірілуінің 2 түрі бар- кростық және резидентті.

2.2.5 Жүйе өлшемдері

НУОЖ үшін маңызды параметр болып орындаушы жүйенің өлшемі- жүйе жиынтығы қосымшасының жұмысы үшін қажетті ең аз жиынтық көлемі болып табылады (ядро, жүйелік модульдер, драйверлер).

2.2.6 Контекстінің ауыстыру уақыты

Нақты уақыттағы операциялық жүйелерде бірнеше оқиғаларды бір мезгілде өндеуге мүмкіндік беретін параллелизм жатыр, сондықтан барлық НУОЖ көп міндетті болып табылады. (көппроцессорлы, көпжелілі). Қатар оқиғаларды өндеуге үстеме шығындарды бағалай білу үшін процестен процеске басқаруды тасымалдау үшін кететін жүйенің уақытын білу керек, яғни контекстінің ауыстыру уақыты.

2.2.7 Таймермен жұмыс істеу құралдары.

Таймермен жұмыс істеу құралдары сияқты құралдар қатты уақыт регламенті бар жүйелер үшін қажет, сондықтан таймермен жұмыс істеу құралдарының дамуы- нақты уақыттағы операциялық жүйелер үшін қажетті атрибут. Бұл құралдар:

- түрлі уақыт аралықтарын белгілеу және өлшеу
- уақыт интервалдарының таусылуына байланысты үзілулерді реттеу;
- бір реттік және циклдік қоңырауларды құруға мүмкіндік береді.

2.3. НУОЖ-нің қолдану аясы

Көптеген уақыт мезетінде НУОЖ-нің негізгі тұтынушылары әскери және ғарыштық облыстар болып келді. Қазір жағдай түгелдей өзгерді, НУОЖ-ні кең таралымды тауарларды кездестіруге болады.

НУОЖ-нің негізгі қолдану аялары:

- әскери және ғарыштық облыстар: борттық және ендірілетін құрылғылар
- басқару және өлшеу жүйелері, радарлар;
- сандық видеожүйелер, симуляторлар
- ракеталар, жергілікті жердің жолын анықтайтын жүйелер.
- өнеркәсіп:
- өнеркәсіпті автоматты басқару жүйелері (ӨАБЖ), (computer – aided manufacturing (САМ)), технологиялық процестерді автоматты басқару жүйелері (ТПАБЖ);
- көлік жасау: симуляторлар, моторды басқару жүйелері, автоматты айқасу;
- энергетика: ақпараттар жиынтығы, құралдар мен мәліметтерді басқару;
- телекоммуникация: коммуникациялық құрылғылар, желілік коммутаторлар, телефон станциялары
- банк құрылғылары (мысалы, көптеген банкоматтарда QNX НУОЖ жұмыс жасайды)
- Кең қолданымды тауарлар:

-ұялы телефондар, мысалы, GSM стандартындағы телефон рSOS НУОЖ-де жұмыс істейді;

- сандық телевизиялық декодерлер;

- сандық телеарна (мультимедиа, видеосерверлер)

- компьютерлік және офистік құрылғылар (принтерлер, көшірмелер), мысалы факстарда VxWorks НУОЖ қолданылады, компакт-дискілерді оқуда VRTX32 НУОЖ қолданылады.

Көп жағдайда НУОЖ бірнеше нұсқаларда болады: жүйенің көлемі бірнеше килобайттан құралғанда толық және қысқартылған болуы мүмкін.

СДЖ арналған бақылау жұмыстары:

1. Нақты уақыттағы орындалу жүйесі деген не? (анықтама бер)
2. Нақты уақыттағы орындалу жүйесіне қандай талаптар қойылады?
3. Нақты уақыттағы жетілдіру жүйесі деген не? (анықтама бер)
4. Нақты уақыттағы жетілдіру жүйесі неден құралады?
5. НУОЖ өзегі қандай қызметтер атқарады?
6. Қорды синхрондау қалай жүзеге асырылады?
7. Міндетаралық айырбас қалай жүзеге асырылады?
8. Мәліметтердің жіктелінуі деген не?
9. Сыртқы құрылғылардың сұраныстарын өңдеу қалай жүзеге асырылады?
10. Ерекше жағдайларды өңдеу не үшін қажет?
11. НУОЖ-дегі қай жоспарлау принципі жиі қолданылады?
12. Жоспарлау кезінде қандай міндеттер жиі атқарылады?
13. Міндеттердің өзара үйлесімділігі қалай атқарылады?
14. Қорларды пайдалану кезінде қандай қиындықтар туады? Критикалық секция деген не?
15. Сыртқы оқиғаларды синхрондау кезінде қандай негізгі беталыстар қолданылады?
16. Уақыт бойынша синхрондау не үшін қолданылады?
17. Процестер мен мәліметтер тасымалын синхрондау құралдарына не жатады?
18. Жүйе реакциясының оқиғаға қатысты уақыты болып не табылады?
19. Жүйе реакциясының уақыты кезінде қандай уақыт аралығы аппаратурамен анықталады?
20. Жүйе реакциясының уақыты кезінде қандай уақыт аралығы операциялық жүйе мен компьютер архитектурасымен анықталады?

Ұсынылатын әдебиеттер:

1. Мартин ДЖ. Программирование для вычислительных систем реального времени. Пер. с англ. Изд-во "Наука", 1975.- 360с.
2. Тенанбаум Э. Современные операционные системы. пер. с англ. 2-е изд. – М.: СПб.: Нижний Новгород: Питер, 2005. – 1037с.
3. Олифер В.Г. Олифер Н.А. Сетевые операционные системы. М.: СПб.: Нижний Новгород: Питер, 2006. – 538с.
4. Грибанов В.П., Дробин С.В., Медведев В.Д. Операционные системы. - М.: Финансы и статистика, 1990. - 239 с.
5. Дейтел Х.М., Чофнес Р.Д. Операционные системы. пер. с англ. – М.: БИНОМ, 2006. – 704с.

16 бөлім. UNIX ОЖ

3. 3 Тақырып . UNIX ОЖ. Жүйенің ерекшеліктеріне жалпы шолу

Дәріс жоспары

16.1 Нақты уақыттағы UNIX-тер.

16.2 UNIX ОЖ

16.2.1 Жүйенің ерекшеліктеріне жалпы шолу

16.2.1.1 Тарихы

16.2.2 Басты түсініктер

16.2.2.1 Тұтынушы

16.2.2.2 Тұтынушы интерфейсі

16.2.2.3 Артықшыланған тұтынушы

16.2.2.4 Бағдарламалар

16.2.2.5 Командалар

16.2.2.6 Процестер

16.2.2.7 Енгізу/шығарудың бағытын өзгерту

16.2.2.8 Аталған бағдарламалық каналдар

16.2.2.9 Қорғаныс принциптері

16.2.2.10 Құрылғыларды басқару

16.2.2.11 Құрылғылардың драйверлері

16.2.2.12 Құрылғылардың ішкі және сыртқы интерфейстері

16.2.2.13 Бағдарламалық ұялар (Sockets)

16.2.2.14 Жойылған процедураларды шақыру

16.3 Жүйенің құрылымы

16.3.1 UNIX ОЖ өзегі

16.3.1.1 Өзекпен өзара әсерлесу принципі

16.3.2 Shell

16.3.3 Командалар

16.3.3.1 Командалар не атқарады?

16.3.3.2 Командаларды қалай орындау қажет?

Нақты уақыттағы UNIX-тер.

Нақты уақыттағы операциялық жүйелер тарихта UNIXтердің бума және өркендеу дәуірінде құрылған, сондықтан олардың көбісі осы операциялық жүйелердің көркем концепциясының осы немесе басқа кірмелерінен тұрады. (тұтынушы интерфейсі, процесс концепциясы және т.б.)

UNIX ОЖ

Жүйенің ерекшеліктеріне жалпы шолу

Тарихы

1965 жылы Bell Telephone Laboratories фирмасы General Electric компаниясы мен Массачусет технологиялық институтының MAC жобасымен күштерін біріктіріп, Multics деп аталған жаңа операциялық жүйе жасауға кірісті. Multics жобасымен жұмыс аяқталған соң Bell Telephone Laboratories фирмасының информатика бойынша зерттеу орталығының қызметкерлері «жеткілікті интерактивті есептеуіш құралдарсыз» қалды. Жетілдіру жағдайларын жақсарту үшін Томпсон және Ричи өздерінің PDP-7 жүйенің жобасын орындады, ол төмендегілерден тұрды:

- UNIX_файлдық жүйесінің бірінші нұсқасы
- Процестерді басқару жүйесі
- шағын утилиттер жиынтығы
- жүйе **жоғары дәрежедегі тілде жазылған**
- тұтынушыға барлық қажетті қызметті ұсынуы мүмкіндік беретін перифериялық құрылғылары бар **қарапайым тұтынушы интерфейсінің болуы**
- Қарапайым программалардан күрделіреу құруға мүмкіндік беретін элементарлы құрылғылардың болуы

- Жұмыста жеңіл және тиімді *иерархиялық файлдық жүйенің болуы*
- Жүйе көп пайдалынымды, көп қызметті болып табылады, әрбір тұтынушы бірмезгілде бірнеше процесті орындай алады.

Барлық UNIX_жүйесі екі бөлікке бөлінеді:

1. Бірінші бөлікті программалар мен сервистік функциялар құрайды, UNIX операциялық ортасын танымал қылады; бұл бөлік тұтынушыға өте оңай қол жеткізеді, ол командалық процессор, хаттар тасымалы, тексті өңдеу процестері, текстік программаларды өңдеу жүйелері сияқты программалардан тұрады.
2. Басқа бөлігі өз қатарына осы программалар мен функцияларды қамтамасыз ететін операциялық жүйені қосады.

Басты түсініктер

Тұтынушы

Есептеуіш файлдық жүйелерде тіркелген адам, сондықтан аталған есептеуіш ат тұтынушы арқылы тіркелген жүйе деп аталады. UNIX ОЖ файлдық жүйе ағаш тәрізді құрылымды болып келеді. Тұтынушы жүйеге енгеннен кейін оның ішіндегі өзінің үй каталогына және барлық каталогтар мен файлдарға шексіз қол жеткізе алады. Тұтынушы үй каталогындағы каталогтар мен файлдарды құру, жою, жаңарту әрекеттерін орындай алады.

Тұтынушы интерфейсі

Тұтынушының UNIX жүйесімен өзара әсерлесуінің дәстүрлі тәсілі командалық тілдерді қолдануға негізделген. (Қазіргі кезде графикалық интерфейстер кеңінен таралған) UNIX ОЖ-нің кез келген командалық интерпретаторының жалпы атауы – Shell (қабықша), себебі кез келген интерпретатор жүйенің өзегінің сыртқы қоршауынан тұрады.

Артықшыланған тұтынушы

UNIX ОЖ-нің өзегі әрбір тұтынушының идентификаторы бойынша (UID- User Identifier) теңестірілуі, тұтынушыға жүйеде тіркелуі кезінде беріледі. НУОЖ-де администратор үшін UID мағынасы ерекше. Осындай UID-сы бар тұтынушы супертұтынушы (superuser) немесе root деп аталады. Оның кез келген файл мен келген программаның орындалуына қол жеткізуге шексіз құқығы бар. Сонымен қатар мұндай тұтынушының жүйені толық бақылайтын мүмкіндігі бар. Ол оны тоқтата алады және жоюы да мүмкін.

Бағдарламалар

UNIX ОЖ бір уақытта қолданбалы программаларды қолданудың операциялық ортасы және жаңа қосымшаларды жетілдіру ортасы болып табылады. Жаңа бағдарламалар түрлі тілде жазылуы мүмкін. (Фортран, Паскаль, Модула, Ада және т.б.) . Бірақ UNIX ОЖ ортасында стандартты программалау тілі С тілі болып табылады. (кейінгі кезден бастап С++ ауыстырылып жүр).

Командалар

Кез келген shell отбасының командалық тілі нақты 3 бөліктен құралады:

1. Текстік қатарларды бақылауға және қарапайым командалардың негізінде күрделі командаларды құруға мүмкіндік беретін **қызметтік конструкциялар**;
2. Командалық тілдің өзіндік интерпретатор арқылы орындалатын **кірме командалар**;
3. Жеке файлдармен ұсынылатын **командалар**.

Процестер

UNIX ОЖ-дегі процес- меншікті адресті виртуальды кеңістікте орындалатын программа. Тұтынушы жүеген енгеннен кейін командалық интерпретатордың программасы орындалатын процес автоматты түрде құрылады.

Енгізу/шығарудың бағытын өзгерту

Механизмнің іске асуы UNIX ОЖ-нің келесі қасиеттеріне негізделген:

- Біріншіден, кез келген енгізу-шығару әлдебір файлдан енгізу және әлдебір файлға шығаруға негізделген;
- Екіншіден, кез келген файлға қол жеткізу дескрипторы арқылы жүзеге асырылады (оң бүтін сан);
- Үшіншіден, әлдебір процесс кезінде іске қосылған программа, пайда болған процестен ашық ашық файлдардың барлық дескрипторларын «мұра ретінде» алады.

Аталған бағдарламалық каналдар

Программалық канал (pipe)- UNIX ОЖ-дегі процесаралық өзара әсерлесудің дәстүрлі құралдарының ең көп тарағандарының бірі.

Программалық каналдың жұмысының басты принципі бір процестің байттық шығаруы буферлеу және FIFO режимінде программалық каналдың мазмұнын басқа процесс арқылы оқуға мүмкіндікті қамтамасыз етуге негізделген. (яғни бәрінен бұрын жазылған байт бірінші оқылады).

Аталмаған программалық канал- ата-процеспен құрылады, процес ұрпақтан мұраға қалады, сондықтан пайда болған процес иерархияларында байланыстың болуын қамтамасыз етеді.

Аталған программалық каналға міндетті түрде әлдебір каталогтың элементі және меншікті і-түйін сәйкес келуі керек. Аталмаған программалық каналдармен салыстырғанда, аталған программалық каналдар кез келген процес байланысы үшін қолданыла алады. (яғни бір иерархия тобына кіретін процестердің болуы міндетті емес).

Қорғаныс принциптері

Тұтынушы жүйеге енгенде login программасы тұтынушы жүйеде тіркелгенін және дұрыс парольді білетіндігін тексереді, жаңа процес тудырады және берілген тұтынушы үшін қажетті shell іске қосады. Бірақ бұдан бұрын login жаңадан құрылған процесс үшін тұтынушы және топ идентификаторын орнатады.

3.2.2.10 Құрылғыларды басқару

UNIX ОЖ сыртқы құрылғыларға қол жеткізу үшін әмбебап файлдың абстракциясы қолданылады.

3.2.2.11 Құрылғылардың драйверлері

Құрылғы драйвері- өзіндік статикалық мәліметтерден тұратын көпкірісті программалық модуль, ол құрылғымен жұмыс істеуді реттей алады, тұтынушымен тапсырыс берілген айырбастарды орындайды, құрылғылардан үзілулерді өңдейді. UNIX ОЖ-де символды, блокты және ағынды драйверлер деп түрленеді.

Символды драйверлер- туынды байттар тізбегін шығару және қабылдауға бағытталған қарапайым және құрылғыларды күтуге арналған драйвер болып табылады. Мұндай драйверлер UNIX-тің өзегі стандартты қызметтің ең аз жиынтығын қолданады, тұтынушы процестің виртуальді кеңістігінен мәліметтер қабылдауға және мәліметтерді осындай виртуальді кеңістікке орналастыруға мүмкіндік беруге негізделген.

Блокты драйверлер- UNIX ОЖ-нің өзегінің блокты айырбасты жүйелік буферлеу мүмкіндіктерін қолданумен жұмыс істейді. Мұндай драйвердің функциялар қатарына UNIX ОЖ өзегінің буфер жүйесіне сәйкес мәліметтер блогын қосу және керек жағдайда буферлік облыстың мазмұнын ашу жатады.

Ағынды драйверлер- күрделі ұйымдасуымен ерекшеленеді. Нақтырақ айтқанда, мұндай драйвер тұтынушы сұранысының көп сатылы өңдеуін қамтамасыз ететін модульдер конвейерінен тұрады. Ағынды драйверлер UNIX ОЖ-нің ортасында негізінен көп деңгейлі, желілі хаттамаларға сәйкес жұмыс істейтін желілі құрылғыларға қол жеткізуді жүзеге асыруға негізделген.

3.2.2.12 Құрылғылардың ішкі және сыртқы интерфейстері

Файлдың түріне қарамастан, (кәдімгі файл, каталог, байланыс және арнай файл) тұтынушылық процесі open, close, read, write сияқты жүйелік шақырулырдан тұратын стандартты интерфейс арқылы файлдармен жұмыс істей алады.

3.2.2.13 Бағдарламалық ұялар (Sockets)

Бағдарламалық ұялар механизмі процесаралық дамыған өзара әсер ретінде жүзеге асырылған. Бұл құрал кез келген процеске желімен қосылған бір компьютерде ме әлде түрлі компьютерде орындалған ба оған қарамастан, кез келген басқа процеспен хат алмасуға мүмкіндік береді.

3.2.2.14 Жойылған процедураларды шақыру

Жойылған процедураларды шақыру механизмінің басты идеялары болып төмендегілер саналады:

(а) Көптеген жағдайларда процестерді өзара әсерлерінің көркем бейнеленген асимметриялы сипаттамасы бар. Бір процес басқа процестен әлдебір қызметті сұрайды, және бұл қызмет орындалмағанша орындауды жалғастырмайды. (процес – клиент сәйкес нәтиже алмағанша)

(б) Тасымалдау қасиеті түрлі компьютерден құралатын операциялық біртекті желілерді құруға мүмкіндік береді, бірақ түрлі архитектурадағы компьютердегі мәліметтердің түрлі көрінісі қиындық тудырады. Сондықтан, RPC-нің екінші идеясы түрлі компьютерде орындалған процестердің өзара әсерлесуі кезінде мәліметтердің форматын өзгертуді автоматтандыру болып табылады.

3.3 Жүйенің құрылымы

UNIX жүйесі 4 басты компоненттерден тұрады:

- өзек
- shell
- commands
- file system

3.3.1 UNIX ОЖ өзегі

UNIX ОЖ –нің дәстүрлі өзегінің машиналы –тәуелді бөлігі келесі компоненттерден тұрады:

- Төмендегі деңгейде жүйенің айналымы және инициализациясы
- Ішкі және сыртқы үзілулерді алғашқы өңдеу
- Жадыны басқару (виртуальды жадының аппаратты тірегінің ерекшеліктеріне қатысты бөлігі)
- Тұтынушы мен өзек режимдері арасындағы контексті процестердің ауысуы
- Құрылғылардың драйверлерінің бөлігінің тұтас платформасының ерекшеліктерімен байланысты

UNIX ОЖ өзегінің басты функцияларына мыналар жатады:

- Жүйенің инициализациясы
- Процестер мен тізбектерді басқару
- Жадыны басқару
- Файлды басқару
- Коммуникациялық құрылғылар
- Программалық интерфейс

3.3.1.1 Өзекпен өзара әсерлесу принципі

UNIX жүйесінде тұтынушылық процестердің орындалуы 2 дәрежеде жүзеге асырылады: тұтынушы дәрежесінде және өзек дәрежесінде.

3.3.2 *Shell*

Shell- операциялық жүйемен байланыс жасауға мүмкіндік беретін программа.

3.3.3. Командалар

Программа- компьютерге арналған инструкциялар жиынтығы. Алдын- ала жарияланусыз компьютермен орындалатын программалар орындаушы программалар немесе командалар деп аталады.

3.3.3.1 Командалар не атқарады?

UNIX жүйесінің сыртқы орта қызметі бойынша категорияларға жіктелген программалар мен аспапты жүйе құрылғыларын құрады. Бұл функциялар:

1. программалық қоршау
2. текстерді өңдеу
3. ақпараттар ұйымы
4. қызмет етуші программалар
5. электроды байланыс

16.3.3.3 Командаларды қалай орындау қажет?

Сіздің сұранысыңыз UNIX жүйесіне түсінікті болу үшін сіз әрбір команданы түзетілген форматта немесе синтаксистік командалық қатарда енгізуіңіз қажет.

СДЖ арналған бақылау жұмыстары:

- 1) UNIX ОЖ-нің басты артықшылықтары қандай?
- 2) UNIX ОЖ қандай 2 үлкен бөліктерге бөлінеді?
- 3) UNIX ОЖ-нің тұтынушылары
- 4) UNIX ОЖ-нің тұтынушы интерфейсі
- 5) UNIX ОЖ-нің артықшыланған тұтынушысы
- 6) UNIX ОЖ –нің программалары
- 7) UNIX ОЖ-нің командалары
- 8) UNIX ОЖ-нің процестері
- 9) UNIX ОЖ-нің енгізу / шығару бағытының өзгеруі
- 10) UNIX ОЖ-нің аталған программалық каналдары
- 11) UNIX ОЖ-нің қорғаныс принциптері
- 12) UNIX ОЖ-нің құрылғыларын басқару
- 13) UNIX ОЖ-нің құрылғылар драйверлері
- 14) UNIX ОЖ-нің құрылғыларының ішкі және сыртқы интерфейстері
- 15) UNIX ОЖ-нің программалық ұялары (Sockets)
- 16) UNIX ОЖ-нің жойылған процедураларды (RPC) шақыруы
- 17) UNIX ОЖ қандай компоненттерден тұрады?
- 18) UNIX ОЖ-нің басты функциялары
- 19) UNIX ОЖ-ің shell басты функциялары
- 20) UNIX ОЖ-нің командаларының басты функциялары
- 21) UNIX ОЖ-нің файлдық жүйесінің басты функциялары
- 22) UNIX ОЖ-нің мобильділігіне қалай қол жеткізуге болады?
- 23) UNIX ОЖ-нің өзегінің машиналы-тәуелді бөлігіне нені жатқызуға болады?
- 24) UNIX ОЖ-нің өзегінің басты функциялары қандай?
- 25) Тұтынушы режимі мен UNIX ОЖ-нің өзегінің режимінің айырмашылықтары неде?
- 26) Өзекпен өзара әсерлесу принциптері. Жүйелік шақырулар деген не?

Ұсынылатын әдебиеттер:

1. А.В.Гордеев, А.Ю.Молчанов. Системное программное обеспечение. — "Питер", 2002. — 736с.
2. Кристиан К. Введение в операционную систему Unix: пер. с англ. — М. Финансы и статистика, 1985. — 360с.
3. Робачевский А.М., Немнюгин С.А., Стесик О.Л. Операционная система Unix. 2-е изд.— СПб.: БХВ – Петербург, 2005. — 635с.

4 тақырып. UNIX ОЖ . Файлдық жүйе. Желілік өзара әсерлесулердің базалық механизмдері. Жадыны басқару.

Дәріс жоспары

ЖҮЙЕНІҢ ҚҰРЫЛЫМЫ (ЖАЛҒАСЫ).

Файлдық жүйе

Кәдімгі жүйелер

Анықтамалар

Арнайы файлдар

Файлдардың түрлі аттармен байланысуы

Виртуальды жадыда бейнелетін файлдар

Файлдарға параллельді қол жеткізу кезіндегі синхрондау

Файлдарды қорғау

Жіктелген файлдық жүйелер

ЖЕЛІЛІК ӨЗАРА ӘСЕРЛЕСУЛЕРДІҢ БАЗАЛЫҚ МЕХАНИЗМДЕРІ

Ағындар (Streams)

TCP/IP хаттамалар ағыны

ЖАДЫНЫ БАСҚАРУ

Виртуальді жады

Жадыны басқарудың аппаратты – тәуелсіз деңгейі

Басты жадының беттік орын басуы және swapping

Жадының жіктелінуі

4.1 Жүйенің құрылымы (жалғасы)

4.1.1. Файлдық жүйе

UNIX жүйесінің басты бірлігі болып саналатын файл кәдімгі файл, анықтама, арнайы файл түрінде болуы мүмкін.

4.1.1.1 . Кәдімгі файлдар

Кәдімгі файлдар белгіліер жиынтығы болып табылады.

4.1.1.2 Анықтамалар

Анықтамалар файлдар мен басқа анықтамалардан тұратын супер-файлдар болып табылады.

4.1.1.3 Арнайы файлдар

Арнайы файлдар терминал, дисктік құрылғылар, байланыс каналдары сияқты физикалық құрылғыларға сәйкес келеді. Арнайы файлдардың 2 түрі бар: **блокты** және **символды**.

Блокты арнайы файлдар өлшемдері 512,1024,4096 және 8192 байттан тұратын мәліметтер байтының блоктарымен айырбас жасалатын сыртқы құрылғылармен бірлеседі.

Символды арнайы файлдар бірдей өлшемдегі мәліметтер блогымен айырбасты талап етпейтін сыртқы құрылғылармен бірігеді.

4.1.1.4. Файлдардың түрлі аттармен байланысуы

UNIX ОЖ –нің файлдық жүйесі түрлі аттары бар бірдей файлдарды байланыстыруға мүмкіндік береді.

4.1.1.5. Виртуальды жадыда бейнелетін файлдар

UNIX ОЖ –нің қазіргі заманғы нұсқаларында кәдімгі файлдарды файлдың мазмұнымен келесі жұмыспен процестердің виртуальді жадысында бейнелеу read, write, lseek сияқты жүйелік шақырулардың көмегімен емес, жадыдан және жазбадан жадыға кәдімгі оқу операторы арқылы бейнелеу пайда болған.

4.1.1.6. Файлдарға параллельді қол жеткізу кезіндегі синхрондау

Синхрондаудың 2 нұсқасы бар:

1. Кідіріспен, яғни блокқа қою талабы бұл талап орындалғанға дейін процесті кейінге қалдырылуы мүмкін.
2. Кідіріссіз, яғни процесс тез арада блокқа қою талабының қанағаттандырылуы туралы жария етіледі немесе берілген уақыт мезетінде оның қанағаттандырылуы мүмкін еместігі туралы жария етіледі.

4.1.1.7 Файлдарды қорғау

UNIX ОЖ –де санцияланбаған қол жеткізуден файлдың қорғанысы 3 фактіге негізделген:

1. Біріншіден, файл тудыратын кез келген процес әлдебір әмбебап жүйеде кейін қайтадан құрылған файл иесінің идентификаторы ретінде саналатын тұтынушы идентификаторына (UID – User Identifier) сәйкес келеді.
2. Екіншіден, файлға әлдебір қол жеткізуге тырысатын әрбір процеспен қос идентификатор ағымдағы идентификаторлар және оның топтары байланысты.
3. Үшіншіден, әрбір файлға оның бейнелеуші – i- түйіні сәйкес келеді.

4.1.1.8 Жіктелген файлдық жүйелер

Жіктелген файлдық жүйелердің басты идеясы басқа компьютерлерде орындалатын процестер үшін локальді файлдық жүйенің файлына бірлесіп қол жеткізуді қамтамасыз ету болып табылады. UNIX ОЖ-нің ортасында барлық белгілі таянулар жойылған файлдық жүйенің локальді файлдық жүйелердің каталогының біреуіне орналастыруға негізделген.

4.2 ***Желілік өзара әсерлесулердің базалық механизмдері***

4.2.1. Ағындар (Streams)

Streams дегеніміз- жүйелік шақырулар мен кішкентай программалардан, өзек қорларынан тұратын байланысқан жалпы қолданыстағы құрылғылар жиынтығы. Негізінен бұл құрылғылар өзек ішіндегі символді енгізу/ шығару стандартты интерфейсін қамтамасыздандырады, сонымен қатар өзек пен сәйкес құрылғы драйверлері арасында, коммуникациялық сервисті жетілдіру және іске асырудың тиімді және дамыған мүмкіндіктерін ұсынады. ***Ағынды модуль*** мәліметтердің ағын бойынша өту қызметінің белгіленген жиынтығын орындайтын мәліметтер өңдеуші болып табылады.

Мәліметтердің тақырыпшадан драйверге немесе модульге, бір модульден келесіне және драйверден немесе модульден ағын тақырыпшасына тасымалдау үшін хаттар механизмі қолданылады. Әрбір хат хаттар блогының жиынтығынан тұрады, ал олардың әрбіреуі :

- мәліметтер блогы
- мәліметтер буферінен тұрады.

4.2.2. TCP/IP хаттамалар ағыны

TCP/IP (Transmission Control Protocol / Internet Protocol) хаттамалар жанұясынан тұрады, олардың басты қызметі – түрлі технологияларға негізделген тиімді компьютерлік желілердің мүмкіндіктерін қамтамасыз ету.

UNIX System V Release 4 TCP/IP хаттамасы TLI (Transport Level Interface – Интерфейс транспортного уровня) қосымша компонент, ағынды модульдер жиынтығы ретінде жүзеге асты. TLI қолданбалы программа мен көлік механизмі арасында интерфейс болып табылады. TLI интерфейсін қолданатын қосымшалардың TCP/IP -ны қолдану мүмкіндігі бар.

4.3 ***Жадыны басқару***

Оперативті жады әрқашан компьютердің критикалық қоры болып келді және солай болып қалады. UNIX ОЖ өзінің өмірін жадыны басқарудың өте қарапайым түрлерін қолданудан бастады. Жүйенің қазіргі заманғы нұсқалары жадыны басқару үшін келесі механизмдерді қолданады.

4.3.1. Виртуальді жады

UNIX ОЖ виртуальді жадының беттік, сегменттік, сегменттік-беттік ұйымдарын қолданады.

Толтыру алгоритмдері

- глобальді
- локальді

болып бөлінеді.

Ең көп таралған дәстүрлі алгоритмдер FIFO (First In First Out) және LRU (Least Recently Used) алгоритмдері болып табылады. Деннингтің алгоритм идесы дегеніміз операциялық жүйенің әрбір уақыт мезетінде процессорға қол жеткізуге бәсекелестік рұқсат етілген барлық процестердің ағымдағы жұмысшы жиынтығын басты жадыда болуын қамтамасыз етуі.

4.3.2 Жадыны басқарудың аппаратты – тәуелсіз деңгейі

UNIX ОЖ виртуальді жадының меншікті бейнеленуін ұйымдастыруға сүйенеді.

Біріншіден, әрбір процестің виртуальді жадысы сегменттердің жиынтығы түрінде ұсынылады.

UNIX ОЖ -нің виртуальді жадысы 5 түрлі сегментке бөлінеді. Сегменттердің 3 түрі әрбір виртуальді жады үшін міндетті , және сегменттер әр түр үшін виртуальді жадыда 1 данадан болады. Олар:

Программалық кодтың сегменті

Мәліметтер сегменті

Стектер сегменті

Бөлінетін сегменттер

Виртуальдыға бейнелетін **файлдар сегменті** әр түрлі бөлінетін сегменттерден тұрады.

Әр сегменттің бейнелеушісі төмендегілерден тұрады:

- сегменттің жеке сипаттамасы, сонымен қатар сегмент басының виртуальді адресі
- Сегменттің байттағы өлшемі
- Берілген сегментпен орындауға болатын операциялар тізімі
- Сегмент мәртебесі
- Сегмент беттерін бейнелеуші кестелерге нұсқаушы және т.б.
- Сонымен қатар, әрбір сегменттің бейнелеушісі берілген виртуальді жадының тура және айнымалы нұсқауларынан және as виртуальді жадының жалпы бейнелеушілеріне нұсқауларынан тұрады.

Осы беттер деңгейінде бейнелеуші құрылымының 2 түрі қабылданған. Физикалық оперативтік жадының әр беті үшін 3 тізімнің біреуінің құрамына кіретін бейнелеушісі бар:

Бірінші тізім

Жаңартуға жіберілмейтін және әлдебір файлдың сыртқы жадысының облысында бейнелетін беттер сипаттаушыларынан тұрады.

Екінші тізім

Еркін беттердің сипаттаушылар тізімі

Үшінші тізім

Жасырын деп аталатын беттердің сипаттаушыларынан тұрады.

4.3.3 Басты жадының беттік орын басуы және swapping

Периодты түрде әрбір процес үшін келесі әрекеттер орындалады:

• Осы процестің виртуальді жадының барлық сегменттерін сипаттау кестелері қарастырылады. Егер сипаттау кестесінің элементі физикалық беттің сипаттаушысына нұсқаудан тұрса онда қаратпа белгілері анықталады.

• Егер белгі орнатылған болса, онда берілген процестің жұмысшы жиынтығына кіретін бет болып табылады, және берілген беттердің санаушы нолге айналады. Егер

белгі орнатылмаса, онда санауышқа бірлік қосылады, ал беттер процесінің жұмысшы жиынтығынан шығатын кандидат мәртебесіне ие болады.

• Егер бұл жағдайда санауыш әлдебір критикалық мағынаға ие болса, беттер процес жұмысшы жиынтығынан шыққан деп саналады, және сыртқы жадыға көшіруге болатын парақтар тізіміне оның бейнелеушісі енгізіледі. Сипаттаушы кесте элементінің қарастыру кезінде олардың әрқайсысында қаратпа белгі өшеді.

4.3.4 Жадының жіктелінуі

Өзек әрқашан осы моменттегі орындалатын процеспен бірге оперативті жадыда орналасады. Компиляция процесі кезінде компилятор-программа айнымалы және ақпаратты құрылғылардың адресі болып табылатын, сонымен қатар нұсқаулықтар мен функциялар адресі болып табылатын адресстер тізбегін реттейді. Компилятор адресстерді физикалық машинада басқа бірде-бір программамен қатар орындалмайтындай етіп виртуальді машина үшін реттейді.

СДЖ арналған бақылау жұмыстары:

1. UNIX ОЖ-де файлдың қандай түрлері бар?
2. Кәдімгі файлдар мен анықтамалар арасында қандай айырмашылық бар?
3. Арнайы файлдар не үшін қолданылады? Арнайы файлдың қандай түрлері бар? Олар қайда қолданылады?
4. UNIX ОЖ-де бірдей файлдар әр түрлі атпен байланыса ала ма?
5. UNIX ОЖ-де виртуальды жадыда файлдарды сипаттау қалай жүргізіледі?
6. UNIX ОЖ-ның бұрынғы нұқаларында файлдарға рұқсатты синхрондау қалай жүргізілген? Қазіргі кезде синхрондаудың қандай нұсқалары қолданылады?
7. UNIX ОЖ-де рұқсат етілмеген қол жеткізуден файлдарды қорғау неге негізделген?
8. Жіктелген файлдық жүйе деген не? Оның кемшіліктері мен артықшылықтары
9. Ағынды драйвердің жұмыс істеу принципі
10. UNIX ОЖ-де виртуальді жадының қандай ұйымдастыру әдістері қолданылады? Оларға сипаттама бер
11. UNIX ОЖ-де виртуальді жадының қандай секіріс (подкачка) қандай ұйымдастыру әдістері қолданылады?
12. Әрбір процесінің виртуальды жадысы қандай сегменттермен берілген? Оларға қысқаша сипаттама бер
13. Аппаратты – тәуелсіз деңгейде сегметті жадының ұйымдастыру әдістері қалай жүзеге асырылады?
14. UNIX ОЖ-де басты жадының беттік орнын басу қалай жүзеге асырылады?

Ұсынылатын әдебиеттер:

1. А.В.Гордеев, А.Ю.Молчанов. Системное программное обеспечение. — "Питер", 2002. — 736с.
2. Кристиан К. Введение в операционную систему Unix: пер. с англ. — М. Финансы и статистика, 1985. — 360с.
3. Робачевский А.М., Немнюгин С.А., Стесик О.Л. Операционная система Unix. 2-е изд.— СПб.: БХВ – Петербург, 2005. — 635с.

5. 5 тақырып. Процестер мен тізбектерді басқару, UNIX ОЖ -дағы енгізу/ шығару. Үзілулер мен ерекше жағдайлар.

Дәріс жоспары

5.1 ПРОЦЕСТЕР МЕН ТІЗБЕКТЕРДІ БАСҚАРУ

5.1.1. Процестердің орындалу ортасы

5.1.1.1 Процестердің тууы

5.1.1.2 Процестердің жоспарлануы

5.1.1.3 Процестерді басқарушы шағын жүйелер

5.1.1.4 Құрастырмалы блок элементтері

5.1.2 *Процестердің тұтынушылық және өзектік құрылымдары*

5.1.3 *Көп қолданбалы режимдердің ұйымдастырылу принциптері*

5.1.4 *Тұтынушы деңгейінде процестерді басқарудың дәстүрлі механизмі*

5.1.5 *Тізбек ұғымы (threads)*

5.2 ЕНГІЗУ / ШЫҒАРУ БУФЕРЛЕУДІҢ ЖҮЙЕЛІК ПРИНЦИПТЕРІ

5.2.1 *Семафорлар*

5.2.2 *Хаттар кезегі*

5.3 ҮЗІЛУЛЕР МЕН ЕРЕКШЕ ЖАҒДАЙЛАР.

5.3.1 *Процессорлық үзілу деңгейі*

5.4 UNIX ОЖ ОРТАСЫН ҚАБЫЛДАЙТЫН КЕЛЕШЕК ОЖ

5.4.1 *Тұтынушының графикалық интерфейс құрылғылары*

5.1 Процестер мен тізбектерді басқару

5.1.1. *Процестердің орындалу ортасы*

5.1.1.1 *Процестердің тууы*

UNIX ОЖ -де жаңа процестің пайда болуы 2 кезеңнен тұрады:

1. Алдымен процесс - әкесінің көшірмесі құрылады, яғни дескриптор, контекст және процесс кейіпі көшіріледі.

2. Одан кейін жаңа процесте кодтың сегменті берілгенінен ауыстырылады.

Қайта құрылған процеске ОЖ бүтін санды идентификатор, жүйенің қызметі барысындағы ең бірегейі тапсырылады.

5.1.1.2 *Процестердің жоспарлануы*

UNIX System V Release 4 жүйесінде басымдықтар мен квантауды қолдануға негізделген ығыстырушы көпміндеттілік іске асырылған.

UNIX System V Release 4 жүйесінде жүйені инсталляциялау кезінде жаңа басымдық класстары қосылуы мүмкін. Қазіргі кезде 3 басымдық кластары бар:

- Нақты уақыт класы
- Жүйелік процестер класы
- Жіктелінетін уақыт процестерінің класы

UNIX бұрынғы нұсқаулармен салыстырғанда процес басымдығы басымдық білдіретін сан үлкен болған сайын жоғары болады. Басымдық мағынасы түрлі кластар үшін әр түрлі анықталады. Жүйелік кластардың процестері белгіленген басымдық стратегияларын қолданады. Жүйелік клас өзек процесі үшін арналған. Процестің басымдық деңгейін өзек белгілейді және ол ешқашан өзгертілмейді. Жүйелік фазаға ауысқан тұтынушылық процес жүйелік басымдықтар класына ауыспайды.

Нақты уақыт процестері де белгіленген басымдықтар стратегиясын қолданады, бірақ тұтынушы оларды өзгерте алады. Басымдықтың әр деңгейі үшін өз кванттық уақытының көлемі қабылданған. Процеске берілген квант уақытында процессорды басып алуға рұқсат етіледі, ал ол біткеннен кейін жоспарлаушы процесті орындаудан алады.

5.1.1.3 *Процестерді басқарушы шағын жүйелер*

UNIX System V Release 4 VFS (Virtual File System) виртуальді файлдық жүйе механизмі іске асырылған, ол жүйе өзегіне бір уақытта файлдық жүйенің бірнеше түлерін қабылдайды. VFS механизмі өзек үшін файлдық жүйе туралы әлдебір абстракты көрніс береді, одан әрбір файлдық жүйенің нақты ерекшеліктерін жасырады.

5.1.1.4 *Құрастырмалы блок элементтері*

Бұрын айтылғандай, UNIX жүйенің жетілдіру концепциясы элементтерінен құралған ОЖ құруға негізделген, олар тұтынушыға күрделілеу программаларды құру кезінде

құрастырмалы блоктар ретінде шығатын шағын программалық модульдер құруға мүмкіндік береді.

Shell командалық процессормен жұмыс істегенде тұтынушылар тап болатын *элементтердің бірі енгізу/шығаруға арналған мүмкіндік* болып табылады.

Екінші құрылымды элемент канал болып саналады, орындалуы оқу мен жазу операцияларымен байланысты процестер арасындағы айырбасты қамтамасыз ететін механизм.

5.1.2 Процестердің тұтынушылық және өзектік құрылымдары

Әрбір процеске контекст сәйкес келеді. Бұл контекст құрамында –

1. тұтынушылық адресті кеңістіктің құрамы – *тұтынушылық контекст* (яғни, виртуальды жадыда бейнеленетін программалық код, мәліметтер, стектің, бөлінетін сегменттер және сегменттік файлдар құрамы)

2. аппаратты тіркегіштің құрамы- *тіркегіш –контекст* (команда санауышының тіркегіші, процессор күйінің тіркегіші, стек көрсеткішінің тіркегіші, жалпы мақсаттағы тіркегіш)

3. Процеспен байланысты берілген өзектің құрылымы – *жүйелік деңгейдегі контекст*.

Процестік контекстінің динамикалық бөлігі -өзек режимінде процестің орындалуы кезінде қолданылатын бір немесе бірнеше стектер. Процестің ядролық стектерінің саны нақты аппаратурамен ұсынылады. Үзілу деңгейінің сандарына сәйкес келеді.

5.1.3 Көп қолданбалы режимдердің ұйымдастырылу принциптері

Бөлу уақыт жүйелерінде кең таралған жоспарлау алгоритмі сақиналық режим (round robin) болып табылады. Уақыттың бөліну кестесі басымдықты процестер арасында жалпы алғанда төмендегідей белгіленеді. Орындалуға дайын процестер процессорға кезекке өз басымдықтарының азаюына байланысты тұрады. Егер кейбір процесс өзінің процесаралық уақыт квантын орындап болса, бірақ әлі орындауға дайын болса, онда әрбір төмендеу басымдықты процес алдына, бірақ сол басымдықты әрбір процестен кейін кезекке тұрады. Егер кейбір процес белсендірілсе, онда ол сол басымдықты процестен кейін кезекке қойылады.

UNIX ОЖ -нің дәстүрлі шешімі динамикалық өзгермелі басымдығын қолдануға негізделген.

5.1.4 Тұтынушы деңгейінде процестерді басқарудың дәстүрлі механизмі

UNIX ОЖ -нің процестерді басқарудың 2 мүмкіндігі бар – командалық тілді қолдану (осы не басқа Shell нұсқасы) және ОЖ-нің өзегінің жүйелік шақырулары тікелей қолданумен программалар тілін қолдану.

5.1.5 Тізбек ұғымы (threads)

«Тізбек» (threads) - әлдебір процестің контекстінде орындалатын басқарудың тәуелсіз ағыны. Нақтырақ айтқанда, процес контексті ұғымы келесі түрде өзгереді. Басқару ағынына жатпайтынның барлығы (виртуальді жады, ашық файл дескрипторы) жалпы процес контекстінде қалады.

5.2 Енгізу / шығару буферлеудің жүйелік принциптері

UNIX ОЖ негізінен енгізу/шығару ұйымының дәстүрлі 3 түрі бар , оған сәйкес 3 драйвер түрі бар.

Блокты енгізу/шығару нақты түрде базалық деңгейде құрылымды каталогтар мен файлдық жүйенің кәдімгі файлдарымен жұмыс істеуге арналған.

Символды енгізу/шығару тұтынушының адресті кеңістігі мен сәйкес құрылғы арасындағы айырбастың тура орындалуына арналған. Барлық далпы өзекті символды драйвер үшін ортақ тұтынушылық және ядролық адресті кеңістіктер арасында мәліметтерді көшіру қызметін қамтамасыз етеді.

Ағынды енгізу/шығару смволды енгізу/шығаруға ұқсас , бірақ аралық өнеуші модульдер ағынына қосу мүмкіндігіне қарай тиімдіреу болады.

5.2.1 Семафорлар

UNIX ОЖ -де семафорлар келесі элементтерден тұрады:

- семафордың мәні
- семаформен соңғы хронологиялық жұмыс жасаған процесс идентификаторы
- семафор маңызы артқанын тосатын процестер саны
- семафордың нөлдік маңызын күтуші процестер саны

5.2.2 Хаттар кезегі

Хаттарды процестер арасында тасымалдау мүмкіндігін қамтамасыз ету үшін бұл механизм төмендегі жүйелік шақыруларды ұстанылады.

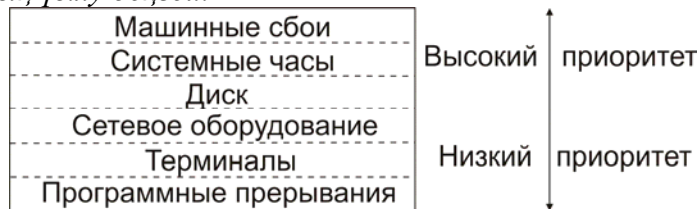
Өзек берілген хатты берілген хаттар кезегіне сәтті қою үшін келесі шарттар орындалуы қажет:

- Қарастырылған процесс берілген хаттар кезегіне жазылу бойынша сәйкес құқықтарға ие болу керек.
- Хат ұзындығы жүйедегі бекітілген жоғарғы шектен асып түспеуі қажет.
- Хаттардың жалпы ұзындығы (қайта жіберілгенмен қоса) бекітілген шектен асып түспеуі қажет.
- Хатта көрсетілген хаттың түрі оң бүтін сан болуы керек.

5.3 Үзілулер мен ерекше жағдайлар

Ерекше жағдайлар процесспен шақырылатын, адресация, артықшыланған командалардың тапсырмалары, нөлге бөлу және т.б. сияқты жоспарланбаған оқиғалардың пайда болуымен тығыз байланысты. Олардың процеске қатысты сыртқы оқиғалармен пайда болатын үзілулерден айырмашылығы бар. Ерекше жағдайлар команданың орындалуының дәл ортасында пайда болады, жүйе ерекше жағдайларды өңдеп, команданы қайта құруға тырысады, үзілулер 2 командалар арасында пайда болады деп есептелінеді, оған қарамастан, жүйе үзілулерді өңдеуден кейін келесі командадан кейін процестің орындалуын жалғастырады. Үзілулер мен ерекше жағдайларды өңдеу үшін UNIX жүйесінде бірдей механизм қолданылады.

5.3.1. Процессорлық үзілу деңгейі



2 сурет. Үзілулердің стандартты деңгейлері

5.4 Unix ОЖ ортасын қабылдайтын келешек ОЖ

Микроядро-модульді және айнымалы кеңейтулер үшін негіз болатын ОЖ-ң ең аз серіппелі бөлігі. Көріп отырғанымыздай, келешек ұрпақтың ОЖ көбісінде микроядролар болады.

Микроядро- ұғымын кең қолданылымға Next компаниясы енгізді, оның ОЖ Mach микроядросын қолданды.

Келесі микроядролық ОЖ Microsoft компаниясының Windows NT болды.

Кейінірек, ОЖ-ң микроядролық архитектуралары Novell/USL, Open Software Foundation (OSF), IBM, Apple және т.б. компаниялармен ұсынылды. Монолитті жүйелерден микроядролық жүйелерге ауысу тенденциясы анықталды.

5.4.1. Тұтынушының графикалық интерфейс құрылғылары

UNIX ОЖ -ң қазіргі заманғы барлық нұсқаларында тұтынушының графикалық интерфейсі жүйемен бірге ұстанылады, тұтынушыға программамен жасалатын графикалық интерфейс жасау үшін аспапты құрылғылар қолданылады.

СДЖ арналған бақылау жұмыстары:

1. UNIX ОЖ -де процестердің тууы қалай іске асырылады?
2. UNIX ОЖ -де басымдықтың неше және қандай түрлері бар?
3. UNIX ОЖ -де құрастырмалы блоктардың қандай элементтері қолданылады?
4. Тұтынушылық және ядролық құрамды процестер
5. UNIX ОЖ -де процестерді қандай жоспарлау алгоритмі қолданылады?
6. Тұтынушы деңгейінде процестерді басқару механизмінің дәстүрлі түрі бар?
7. UNIX ОЖ -дегі тізбек ұғымы
8. UNIX ОЖ -де енгізу/шығару ұйымының қандай басты түрлері бар?
9. UNIX ОЖ -дегі семафорлар
10. UNIX ОЖ -дегі хаттар кезегі
11. UNIX ОЖ -дегі үзілулер мен ерекше жағдайлар
12. UNIX ОЖ-нің ортасын қабылдайтын келешек ОЖ

Ұсынылатын әдебиеттер:

1. А.В.Гордеев, А.Ю.Молчанов. Системное программное обеспечение. — "Питер", 2002. — 736с.
2. Кристиан К. Введение в операционную систему Unix: пер. с англ. — М. Финансы и статистика, 1985. — 360с.
3. Робачевский А.М., Немнюгин С.А., Стесик О.Л. Операционная система Unix. 2-е изд.— СПб.: БХВ – Петербург, 2005. — 635с.

6 тақырып. Windows NT. Негізгі сипаттамалары. НУОЖ секілді Windows NT ОЖ. ОЖ үшін нақты уақыт кеңейтілулері. Windows NT.

Дәріс жоспары

6.1 Windows NT. Негізгі сипаттамалары.

6.1.1. Құрылымы: NT executive және қорғалған жүйелер

6.1.2. Windows NT тұжырымдамалары

6.1.2.1 Көптеген қолданбалы орталар

6.1.2.2. Объекті – бейімделген бағыт

6.1.2.3. Процестер және тізбектер

6.1.2.4 Процестер мен тізбектерді жоспарлау алгоритмі

6.1.2.5. NT үзілулерін өңдеу

6.1.2.6 Желілік құрылғылар

6.2. Windows NT ОЖ НУОЖ дәрежесінде

6.2.1 НУОЖ ретінде Windows NT-ті қолдану мүмкіндігі

6.2.1.1. НУОЖ негізінде Windows NT қолдану үшін Microsoft аргументі

6.2.1.2. НУОЖ негізінде Windows NT қолдануға қарсы аргумент

6.2.2. Windows NT ОЖ үшін нақты уақыт кеңейтілулері

6.2.2.1 Windows NT жүйе үшін нақты уақыт кеңейтілуі туралы толығырақ

6.1 Windows NT. Негізгі сипаттамалары.

1993 жылдың ортасынан бастап, Microsoft «жаңа технологиялар» (New Technology -NT) деп аталатын жаңа ОЖ шығара бастады.

Windows NT ОЖ басынан бастап, қазіргі заманғы ОЖ-ге ұсынылатын барлық талаптармен жобалана бастады:

- кеңейтілуі
- тасымалдылық
- сенімділік
- үйлесімділік

- өнімділік

Бұл қасиеттер *клиент – сервер, микроядролар, объектілер* сияқты құрылымды жобалаулар технологияларын қолдану арқылы қол жеткізілді.

Windows NT-да *ығыстырушы көп міндетті механизм* қолданылады (preemptive multitasking). Тізбекті басқару үшін Windows NT Server басымдықты механизмді пайдаланады, Windows NT есептеуіш процестің симметриялы көп процессорлы ұйымды пайдаланады, оған сәйкес ОЖ процессорлар арасындағы жадыны бөліп кез келген бос процессор мен барлық процессорларда бір мезгілде орындалады. Windows NT Server 16 параллельді процессорды ұстанады.

Енгізу/шығару құрылғыларын басқару кезінде Windows NT асинхронды жолды қолданады.

6.1.1. Құрылымы: NT executive және қорғалған жүйелер

Windows NT құрылымын жасағанда жеткілікті дәрежеде микроядро тұжырымдамасы қолданылады. Осы идеяға сай ОЖ бірнеше жүйелерге бөлінеді, олардың әрқайсысы жеке сервисті қызметтер жиынтығын орындайды, мысалы жады сервисі, процестерді құру сервисі, процестерді жоспарлау сервисі. Мұның қалай жұмыс жасайтынын толығырақ қарастырайық.

Құрылымы бойынша Windows NT 2 бөлік ретінде ұсынылады: (1 сурет)

1. тұтынушы режимінде жұмыс жасайтын ОЖ бөлігі
2. ядро режимінде жұмыс жасайтын ОЖ бөлігі

Ядро режимінде жұмыс істейтін ОЖ бөлігі executive – орындаушы бөлігі деп аталады. Ол виртуальды жадыны, объектілерді (қорларды), енгізу/шығару және файлдық жүйені (сонымен қатар желілік қорлар), процестердің өзара әсерлесуі және қауіпсіздік жүйелерін басқаратын компоненттерден тұрады. Бұл компоненттер модульаралық байланыс арқылы өзара байланысады. Әрбір компонент басқаларын мамандандырылған ішкі процедуралар жиынтығы арқылы шақырады.

Windows NT-ның екінші тұтынушы режимінде жұмыс істейтін бөлігі- қорғалған жүйелер деп аталатын серверлерден құралады. Windows NT серверлері қорғалған жүйелер деп аталады, өйткені олардың әрқайсысы жеке жадысы басқа процестерден NT executive виртуальды жадыны басқару жүйесімен бөлінген процесте орындалады.

Қорғалған жүйенің тірегін Windows NT executive орындаушы бөлігі қамтамасыз етеді, ол ядро кеңістігінде жұмыс істейді және дискке ешқашан көшірілмейді. Оның құраушы бөліктері:

Объект менеджері

Қауіпсіздік мониторы

Процестер менеджері

Виртуальді жады менеджері

Енгізу/шығару жүйесі

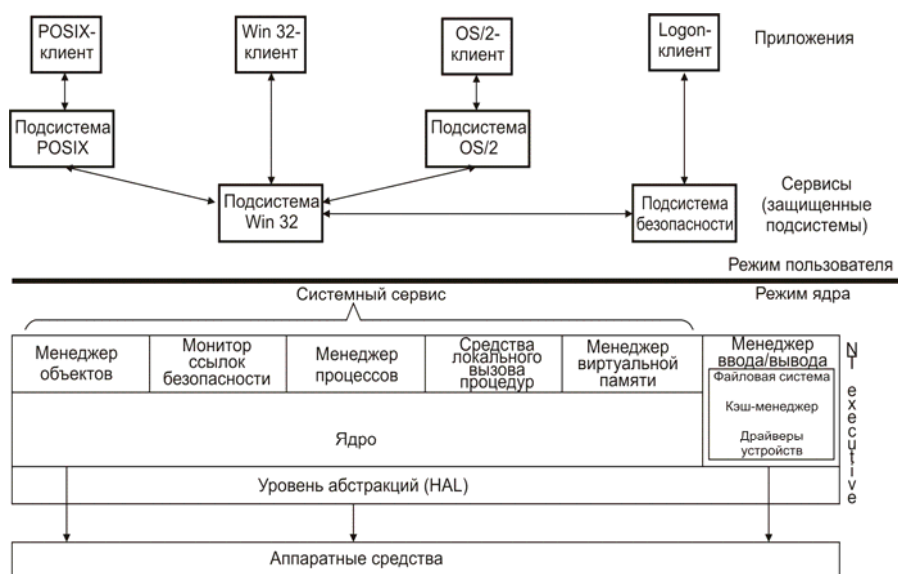
Орындаушы бөлігі өз кезегінде, NT ядросы (оны микроядро деуге болады) ұсынатын төменгі деңгей қызметіне негізделген.

Ядроның негізгі қызметі:

- процестерді жоспарлау
- үзілулер мен ерекше жағдайларды өңдеу
- көп процессорлы жүйелер үшін процессорларды синхрондау
- ақаудан кейін жүйені қалпына келтіру

Ядро артықшыланған режимде жұмыс істейді және ешқашан жадыдан өшірілмейді. Ядроға үзілу арқылы ғана көңіл аударуға болады. Ядро машиналы-тәуелді процедуралардың көп бөлігі бір жерде аппаратты абстракция деңгейінде (Hardware Abstraction Level HAL) орналасқан. HAL NT executive және аппаратты қамтамасыз ету арасында орналасады және жүйеден үзілу бақылаушылар, енгізу/шығару интерфейстері және процессорлар арасындағы өзара әсерлесу механизмдерін жасайды. Мұндай шешім

Windows NT-ға бір платформадан екіншісіне HAL қабатын ғана айырбастау жолымен өте оңай тасымалдауға мүмкіндік береді.



3 сурет Windows NT-ның құрылымы

6.1.2. Windows NT тұжырымдамалары

6.1.2.1 Көптеген қолданбалы орталар

Windows NT операциялық жүйені бес қолданбалы ортасын пайдаланады: MS – DOS, 16- разрядты Windows, OS/2 1.x, POSIX, 32 разрядты Windows (Win32).

6.1.2.2. Объекті – бейімделген бағыт

NT толығымен объекті – бейімделгіш болмағанымен, оның негізінде объектілер жатыр. Атаулардың бірігіп қолдануының және жүйелік қорлардың есептік біркелкі формасы, қауіпсіздігін қамтамасыз етудің қарапайым және арзан әдісі және модификациялық барлық осы қасиеттерге объекті моделді қол жеткізуге болады. Әрбір объект 2 бөліктен тұрады:

- объектінің тақырыбы
- объектінің денесі

олар объектінің стандартты және ауыспалы деректерінен тұрады. Объект менеджері объектінің тақырыбымен жұмыс жасайды, ал қалған компоненттер executive өздері жасаған объекті түрінің денелерімен жұмыс істейді.

Объектінің тақырыбының кез келген түрінде төмендегі белгілер болады:

- аты
- каталог
- қауіпсіздік дескрипторы
- қорларды қолдануға квоталар
- ашық сипаттаушылар есептегіші
- ашық сипаттаушылардың деректер қоры
- тұрақты/уақытша белгілер
- тұтынушы/ядро режимі
- объектілердің түріне нұсқаушы

Объектінің тақырыбынан басқа әрбір объект денеден тұрады. Оның форматы және мазмұны осы объектінің түрімен анықталады. Барлық объектілерде бірдей формат денесі болады. Объектіні құру кезінде орындаушы бөлігі осы түрдегі объектінің денедегі деректерге операция жасау.

6.1.2.3. Процестер және тізбектер

Windows NT-ның процестері келесі сипаттардан тұрады:

- Windows NT-ның процестері объектілер формаларында іске асырылған және оларға қол жеткізу объектінің қызметтері арқылы жүзеге асырылады.
- Windows NT-ның процестері көп тізбекті ұйымнан тұрады.
- Объекті – процестер объекті тізбектер сияқты синхрондаудың орнатылған құрылғыларынан тұрады.
- Windows NT-ның процес менеджері –процестер арасында, «әке- ұрпақ» секілді қатынастарды қолдамайды.

Кез келген жүйеде «процесс» ұғымы келесілерден тұрады:

- Орындалушы код
- Виртуальды адрес жиынтығынан тұратын және процесті қолданатын меншікті адресі кеңістік
- Операциялық жүйе процеске тағайындалған файлдар, семафорлар сияқты жүйе қорлары
- Ең болмағанда бір орындалатын тізбектің болуы

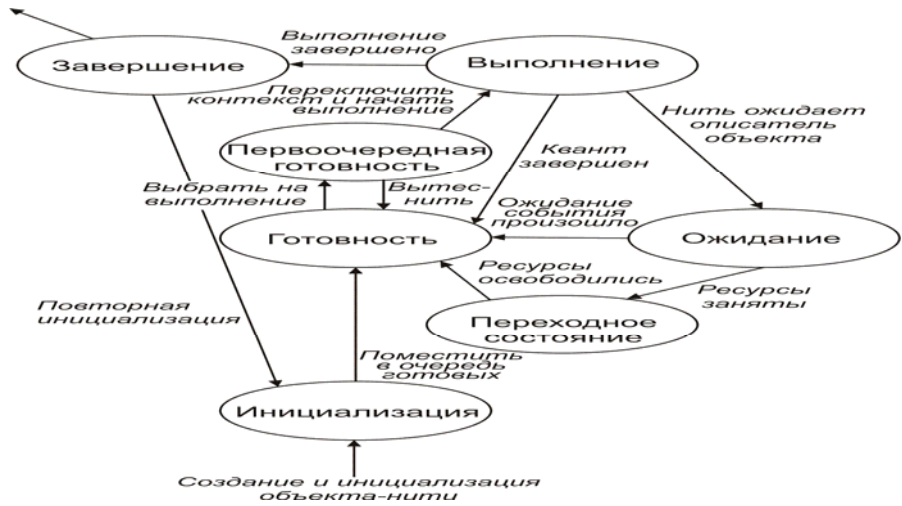
6.1.2.4 Процестер мен тізбектерді жоспарлау алгоритмі

Windows NT-да ығыстырушы көпміндеттілік іске асырылған. Оның негізінде операциялық жүйе тізбектің процесорларды қашан босататынын тоспайды, ол берілген уақытты кетріп болғаннан кейін еріксіз оны орындаудан дайын кезекте жоғары басымдықты тізбек пайда болса босатады

Windows NT 2 класқа бөлінген 32 деңгейлі приоритеттен тұрады:

- Нақты уақыт класы
- Айнымалы приоритет класы

16 –дан 31 диапазон аралығында орналасқан приоритеттері бар нақты уақыт тізбектері приоритетті процес болып табылады және уақыт бойынша критикалық міндеттің орындалуы үшін қолданылады.



6.1.2.5. NT үзілулерін өңдеу

Өңдеу екі кезеңде жүргізіледі:

- Алдымен ең қысқа үзілудің қамтамасыз ету бағдарламасы орындалады (ISR)
- Кейін жұмыс DPC – кейін қалдырылған шақыру процедураларымен аяқталады.

Келесі оқиғалар ағыны туындайды:

- Үзілу пайда болады
- Процессор PC, SP сақтайды және диспетчерді шақырады
- ОЖ контексті сақтайды және ISR-ды шақырады
- ISR-де критикалық жұмыс орындалады (оқу/жазу аппарат регистрлерін)
- DPC кезекке қойылады

- ОЖ контексті қалпына келтіреді
- Процессор PC, SP қалпына келтіреді
- Кезекте тұрған DPC DISPATCH LEVEL приоритеті деңгейінде орындалады
- Барлық DPC аяқталғаннан кейін ОЖ қосымшаның орындалуына көшеді

6.1.2.6 Желілік құрылғылар

Windows NT желілік құрылғыларды жаңарту негізінде келесі қасиеттерге ие болады:

- Драйверлер деңгейінде **ендірілген**. Бұл қасиет жылдам әсер етуді қамтамасыз етеді.
- **Ашықтық** – динамикалық – жүктеу-жүктелу жеңілдігіне негізделген.
- **RPC –ның болуы**, аталған конвейер мен пошталық жәшіктерге жіктелген қосымшалар үшін арналған
- Корпорация шеңберінде желіні құруға рұқсат ететін **қосымша желілік құрылғылардың болуы**. Қосымша қауіпсіздік құрылғылары орталықтандырылған административтік қарсы тұру қабілеті.

6.2. Windows NT ОЖ НУОЖ дәрежесінде

6.2.1 НУОЖ ретінде Windows NT-ті қолдану мүмкіндігі

6.2.1.1. НУОЖ негізінде Windows NT қолдану үшін Microsoft аргументі

- *Жүйенің көпміндеттілік*
- *Көп процессорлықты қолдау*
- *Preemption міндеттері*
- *Preemption үзілулері және оларда жасыру мүмкіндігі*
- *Асинхронды енгізу/шығару*
- *HAL интерфейсі арқылы құрылғыға тура қол жеткізу*
- *Арнайы приоритет сызбалары*

6.2.1.2. НУОЖ негізінде Windows NT қолдануға қарсы аргумент

- *Жүйенің ядросы Preemptive емес*
- *DPC механизм кемшіліктері*
- *Нақты уақыт класындағы приоритеттің аз көлемі*
- *Приоритеттің инверсия мәселелері шешілмеген*
- *Жоғары приоритетті міндеттер төменгі приоритеттілермен блокка қойылуы мүмкін*
- *VirtualLock шақыруына қарамастан белсенді емес процестің барлық беттері (мысалы, мәліметтерді күтетін) дискіге көшірілуі мүмкін*
- *Windows NT үшін жүйелік шақырулар уақыты мен үзілулердің блокка қойылу уақыты келтірілмейді*
- *Жүйені дисплей мен перне тақтасыз қолдану мүмкін емес*
- *Жүйе НУОЖ үшін жадыға үлкен көлемді сұраныстарды ұсынады*

Осы кемшіліктерді жою үшін компаниялар программалық (аппараттық) құрылғыларды ұсынады.

6.2.2. Windows NT ОЖ үшін нақты уақыт кеңейтілулері

6.2.2.1 Windows NT жүйе үшін нақты уақыт кеңейтілуі туралы толығырақ

Кеңейтілуді жасаушылар екі жолмен жүрді:

1. НУОЖ ядросын Windows NT ядросына қосымша ретінде қолданды. Бұл «LP Elektroniks» және «Radisy» ф ирмасының шешімі болды. Бірінші жағдайда Windows NT-пен қатар VxWorks ОЖ жұмыс істейді, екінші жағдайда InTime . Сонымен қатар нақты уақыт байланыс қосымшалары мен Windows NT қосымшалары үшін қызметтер жиынтығы ұсынылады.

VenturCom фирмасының нақты уақыт кеңейтілу нұсқалары басқаша болды. Бұл жерде нақты Windows NT-да кідірістер мен тоқтап қалу себептерін зерттеуге және осы себептерді нақты уақыт жүйесі арқылы жою жолымен «интегралдау» тәсілін қолданды. «VenturCom» фирмасының шешімімен Windows NT аппаратты абстракция деңгейінде

жаңарту-драйверлердің аппаратурамен өзара әсерлесетін программалық қабатқа негізделген.

СДЖ арналған бақылау жұмыстары:

1. Windows NT – ның басты сипаттамалары
2. Windows NT-да микроядро концепциясы не арқылы жүзеге асырылады?
3. Windows NT-да қандай 2 режим бар?
4. Windows NT-да сервистер мен серверлер қандай режимдерде орындалады және олардың өзара әсерлесулері қалай орындалады?
5. Серверлердің, сервистердің, енгізу/ шығару менеджерлерінің , микроядро және аппаратты абстракция деңгейлерінің негізгі қызметтері.
6. Windows NT-ның барлық программаларын біртұтас жүйеге біріктіретін басты құрал не?
7. ОЖ-нің қандай қолданбалы орталдарын Windows NT қабылдайды?
8. Объектіленр деген не, олар не үшін қолданылады және олар қандай бөліктерден тұрады?
9. Windows NT-да процестер мен тізбектер қалай жүзеге асырылады?
10. Windows NT-дағы тізбектердің жоспарлау алгоритмдері
11. Windows NT-нің интерактивті есептер үшін қандай жоспарлау жүзеге асырылады?
12. Windows NT-дағы үзілулерді өңдеу
13. Windows NT-дағы желілік құралдар
14. NT НУОЖ ретінде қолдану мүмкіндігі. «Иә» және «жоқ» аргументтері
15. NT НУОЖ ретінде қолдану мүмкіндігінің 2 шешімі

Ұсынылатын әдебиеттер:

А.В.Гордеев, А.Ю.Молчанов. Системное программное обеспечение. — "Питер", 2002. — 736с

7. 7 дәріс. «Классикалық» НУОЖ. Объекті-бейімделген жүйелер.

Дәріс жоспары

7.1. «Классикалық» НУОЖ

7.1.1. QNX ОЖ

7.1.1.1. QNX ОЖ-нің басты сипаттамалары

7.1.1.2. QNX микроядросының архитектурасы

7.1.1.2.1. Шын ядро

7.1.1.2.2. Жүйелік процестер

7.1.1.2.3. Құрылғылар драйверлері

7.1.1.2.4. Процесаралық байланыс (IPC)

7.1.1.3. QNX хаттар тасымалының ОЖ ретінде

7.1.1.4. QNX желі ретінде

7.1.1.4.1. Бірлік компьютер моделі

7.1.1.4.2. Желінің тиімді ұйымдастырылуы

7.1.2. рSOS

7.1.3. VxWorks

7.2 Объекті- бейімделген жүйелер

7.2.1. OS-9

7.2.1.1. OS-9 басты сипаттамалары

7.2.2. Linux ОЖ үшін нақты уақыт кенейтілулері

7.2.3.1. Linux-те ядролардың жүктелуші модульдері

7.2.3.2. Linux негізіндегі жүйелер

7.2.3.3. Linux жүйесінің даму бағыттары
7.2.4. Soft Kernel

7.1. «Классикалық» НУОЖ

Программалауға сүйенетін классикалық процедураларға негізделген жүйелерді қарастырайық.

7.1.1. QNX ОЖ

7.1.1.1. QNX ОЖ-нің басты сипаттамалары

QNX жүйесін QNX SoftWare Systems (USA) фирмасы өндіреді. Басты сипаттамалары:

3. Классикалық жүйелер түрі: self-hosted
 4. Архитектура: Микроядро негізінде
 5. Стандарт: POSIX 1003
 6. Қасиеттері НУОЖ-нікіндей:
 - көп міндеттілік: POSIX 1003
 - көп процессорлық
 - 32 деңгейлі приоритет
 - жоспарлау: FIFO, round robin, адаптивтік
 - preemptive ядросы
 7. ОЖ жасау (host): UNIX/Windows.
 8. Процессорлар (target): Intel 80x86
 9. Байланыс сызықтары host/target: Ethernet, Arcnet, Serial, Token Ring.
 10. Ең аз өлшемі: 60 Kb
 11. Синхрондау және өзара әсерлесу құралдары: POSIX 1003 (семафорларб mutex . . .)
 12. Жасау құралдары:
 - Компилятор C/C++, түзеткіш, QNX анализдеушісі және тәуелсіз жеткізушілерден (мысалы , Watcom/ SyBase)
 - QNX үшін X Windows / Motif
- QNX бұл жоғары модульділік, тиімділік, қарапайымдылық деген атқа 2 іргелі принцип арқылы қол жеткізді:
- Микроядролық архитектуралар
 - Хаттарға негізделген процесаралық байланыс

7.1.1.2. QNX микроядросының архитектурасы

QNX өзара әсерлесуші процестердің тобына жауапты шағын ядродан құралады. Келесі суреттен көріп отырғанымыздай, құрылым иерарихиядан гөрі көбінесе топқа ұқсайды-бірнеше ойыншылар бірдей рангтің бір-бірімен және өзінің ядро «төрттігімен» өзара әсерлеседі.



5. сурет. Жүйелік менеджерлермен суреттелетін QNX микроядросы

Шын ядро

Ядро- кез келген операциялық жүйенің негізі, QNX-те оған 2 қызмет арналған:

- Хаттар тасымалы- микроядро барлық жүйенің барлық процестер арасындағы барлық хаттардың жолын басқарады.
- Доспарлау- бұл микроядроның бөлігі, ол процес өз күшін хат және үзілу нәтижесі ретінде өзгерткен сайын шақырылады.

Жүйелік процестер

QNX конфигурациясы келесі жүйелі процестерден тұрады:

- Процестер менеджері (Proc)
- Файлдық жүйе менеджері (Fsys)
- Құрылғылар менеджері (Dev)
- Желілік менеджер (Net)

Құрылғылар драйверлері

Құрылғылар драйверлері- анықталған аппаратты құралдар үшін талап етілетін ОЖ-ні құралдармен әсерлесуден қорғайтын процестер.

Процесаралық байланыс (IPC)

QNX қарапайым, бірақ мықты IPC мүмкіндіктерін қамтамасыз етеді.

7.1.1.3. QNX хаттар тасымалының ОЖ ретінде

QNX-те хат дегеніміз- бір процестен келесіне өтетін байттар пакеті. QNX-те хаттың құрамына ешқандай арнайы көңіл бөлінбейді-хаттағы деректердің тек хатты жіберуші мен қабылдаушыға маңызы зор.

7.1.1.4. QNX желі ретінде

Ең қарапайым формада локальді желі бірнеше біріккен компьютерлер ортасындағы файлдар мен перифериялық құралдарды бірігіп қолдану механизм арқылы қамтамасыз етіледі. Процестер желінің барлық жерінде осы әдіспен байланысады.

Бірлік компьютер моделі

QNX негізінен глобальді желілік ОЖ ретінде жасалған. QNX желісі әмбебап микрокомандалар жиынтығынан компьютерлерді қабылдайды.

Желінің тиімді ұйымдастырылуы

QNX-тегі желіде әрбір түйінге оның номері беріледі, кейін ол оның идентификаторы болады. Бұл номер QNX желілік немесе жеке ОЖ ретінде орындалатынын анықтайды.

7.1.2. pSOS

pSOS жүйесі Integrated Systems (Santa Clara, USA) фирмасымен өндіріледі. 2000 жылы ақпанда фирманы Wind River Systems (Alameda, CA, USA) компаниясы алды.

Басты сипаттамалары:

1. Түрі: host/target
2. Архитектура: Микроядро негізінде
3. Стандарт: өзіндік
4. Қасиеттері НУОЖ-нікіндей:
 - көп міндеттілік
 - көп процессорлық
 - 255 деңгейлі приоритет
 - жоспарлау: приоритетті
 - preemptive ядросы
5. ОЖ жасау (host): UNIX/Windows.
6. Процессорлар (target): Motorola 68xxx, Intel 80x86, ARM, MIPS, Power PC
7. Байланыс сызықтары host/target: тізбекті канал, Ethernet.
8. Ең аз өлшемі: 15 Kb
9. Синхрондау және өзара әсерлесу құралдары: семафорлар, mutex , оқиғалар және т.б.
10. Жасау құралдары:

C/C++/ADA жасаудың интегралданған ортасы

7.1.3. VxWorks

VxWorks жүйесі Wind River Systems (Alameda, CA, USA) фирмасымен өндіріледі.

Басты сипаттамалары:

1. Түрі: host/target
2. Архитектура: монолитті
3. Стандарт: өзіндік және POSIX 1003
4. Қасиеттері НУОЖ-нікіндей:
 - көп міндеттілік
 - көп процессорлық
 - 256 деңгейлі приоритет
 - жоспарлау: приоритетті
 - preemptive ядросы
5. ОЖ жасау (host): UNIX/Windows.
6. Процессорлар (target): Motorola 68xxx, Intel 80x86, Power PC , SPARC, Alpha, ARM, MIPS.
7. Байланыс сызықтары host/target: тізбекті канал, Ethernet, VME шинасы
8. Ең аз өлшемі: 5.8 Kb
9. Синхрондау және өзара әсерлесу құралдары: семафорлар POSIX 1003, кезектер, белгілер...
10. Жасау құралдары:
 - TORNADO C/C++ жасаудың интегралданған ортасы
 - UNIX үшін эмуляторы VxSIM
 - WindView есептердің күйінің графикалық визуализаторы

7.2 Объекті- бейімделген жүйелер

7.2.1. OS-9

7.2.1.1. OS-9 басты сипаттамалары

1. Түрі: host/target
2. Архитектура: микроядро негізінде
3. Стандарт: өзіндік , UNIX –қа ұқсас шақырулар
4. Қасиеттері НУОЖ-нікіндей:
 - көп міндеттілік
 - көп процессорлық
 - 255 деңгейлі приоритет
 - жоспарлау: приоритетті, FIFO, арнайы жоспарлау механизмі
 - preemptive ядросы
5. ОЖ жасау (host): UNIX/Windows.
6. Процессорлар (target): Motorola 68xxx, Intel 80x86, Power PC , ARM, MIPS.
7. Байланыс сызықтары host/target: тізбекті канал, Ethernet
8. Ең аз өлшемі: 16 Kb
9. Синхрондау және өзара әсерлесу құралдары: бөлінетін жады, семафорлар, оқиғалар, белгілер
10. Жасау құралдары:
 - Hawk - C/C++ жасаудың интегралданған ортасы
 - PersonalJava – Java виртуальді машинасы

7.2.3. Linux ОЖ үшін нақты уақыт кеңейтілулері

11. Linux негізіндегі жүйелерді қарастырамыз- UNIX жүйесімен еркін таралады. Ол үстел компьютерлерінде өзінің тегін және сапалы күшінде тараған. Intel 80x86 машиналарында процессор мен пайда болғаннан кейін қазір Alpha , SPARC, Power PC , ARM, Motorola 68xxx, MIPS процессорларын қабылдайды .

Linux жүйесінің нақты уақыт талаптарына 3 бағыт жатады:

1. Нақты уақыт жүйелеріне қатысты POSIX стандартын қолдау. POSIX 1003. 1c (thread міндеттермен жұмыс) стандарты қабылданды, POSIX 1003. 1b стандартының бөлігі ғана қабылданды: жадыны басқару механизмі және есептерді жоспарлау механизмі жүзеге асырылады, таймерлермен жұмыс істеу механизмдері, белгілер, POSIX семафорлары, хаттар кезегі әлі жүзеге асырылмаған.

2. Арнайы құрылғыны қолдау, олардың ең маңыздысы VME шинасы. VME. PCI мостын қолдау бар. ПЕСҚ-нан Linux –тың орындалуын қамтамасыз ету бойынша жұмыстар жүргізіліп жатыр. Нақты уақыт жүйелері жүйе таймерлерінің жоғарылауы маңызды.

3. Жүйе ядросы үшін preemption механизмін іске асыру. Бұл механизм бір жағынан жүйенің нақты уақыт жүйесі деп атау үшін қажет, екінші жағынан, ол жүзеге асырылу үшін өте күрделі. Linux басқа UNIX жүйелері сияқты көпке дейін жүйе өзегіне кіруге қарсы болды, және preemptive болмайды.

Linux ядросы үшін preemption жүзеге асырудың бірнеше жобалары бар. Есептерді шешу тәсілдері бойынша оларды екі топқа бөлуге болады.

1. preemption механизмі жүйе ядросын көшіру арқылы жүзеге асырылады. Осы жолда ең сапалы нәтижелерге қол жеткізуге болады, бірақ қазіргі уақытта сапалы нәтижелер жоқ өйткені:

А. Ядроның үлкен көлеміне байланысты жұмыстың көлемі өте үлкен

В. Ядроның өзгеру жылдамдығы өте жоғары, өзгерістер нақты уақытқа байланысты емес

2. preemption механизмі микроядроны жазу арқылы жүзеге асырылады, ол үзілулер мен есептерге диспетчер жауап береді. Linux ядросы төменгі приоритетпен жұмыс істейді, мұндай жүйедегі міндеттер екі топқа бөлінген:

А. тек микроядроның басқаруымен жұмыс істейтін процестер, бұл процестер Linux ядросында үзілетіндіктен нақты уақыт талаптарын қанағаттандырады.

В. Linux басқармасымен жұмыс істейтін процестер, сонымен қатар микроядро басқарумен жұмыс істейтін есептер, бірақ Linux функцияларын қолданады; бұл процестер Linux ядросында блокка қойылатындықтан, нақты уақыт талаптарын қанағаттандырмайды.

Мұндай жолдың *кемшілігі* нақты уақыт процестерінің қызметімен қамтамасыз етілетін микроядроны жүзеге асырудың қажеттілігі. Мысалы, нақты уақыт процесі коммуникациялық портпен жұмыс істегісі келсе, (мысалы, СОМ порты), онда бұл порттың драйвері Linux ядросынан микроядроға ауыстырылуы қажет.

7.2.2.1. Linux негізіндегі жүйелер

- RT-Linux
- KURT

7.2.2.2. Linux жүйесінің даму бағыттары

Linux және Hurd

Карнеги-Меллон университетінде Mach жобасы жасалынды- операциялық жүйенің микроядролық архитектурасы. Микроядро процестерді, олардың арасындағы хаттар айырбасын, виртуальді жадыны және құрылғылар драйверлерін басқарады. Операциялық жүйенің қалған бөлігі серверлер түрінде жүзеге асады- тұтынушылық режимде орындалатын программалар.

Сервер тұтынушы үшін жүйенің түрін анықтайды. Жүзеге асқан сервер жүйені UNIX немесе DOS- қа айналдырады.

Микроядро көп процессорлы жүйе жұмысына бейімделген, аппаратты платформаның жоғарғы тәуелсіздігі мен тұтынушы құқығына ие қазіргі заманғы технология.

Hurd, UNIX нұсқасы ретінде Mach микроядросының серверлер түріндегі жүзеге асырылған. Еркін операциялық жүйе Hurd жүйесі жұмыс істеп тұрғанмен, әлі аяқталған

жоқ. Бұл міндеттерді Столлман RISC- машинасының желілік функциялары мен жүйенің нұсқасының іске асырылуы ретінде санайды.

Hurd GNU жобасының аяқталған шағы - UNIX операциялық жүйесінің стандартты еркін ортасын құру – оны FSF қоры құрады. Столлман қазірдің өзінде GNU – Linux операциялық жүйесі толық іске асырылу жобасы бар деп аталады, өйткені Linux құралына GNU жобасының программалық қамтамасыздандырылуы жатады. (трансляторлар, редакторлар, утилиттер және т.б.)

Hurd үшін Linux файлдық жүйесінің серверін жасаушы Луи—Доминик Дюбо (Louis –Dominique Dubeau) былай дейді: « Hurd өте жақсы жобаланған, ол сенімді ақтайды деп ойлаймын. Hurd технологиясын микроядро негізінде Linux жүзеге асыру үшін қолдану – қазіргі заманда ең тиімді шешімдердің бірі».

Сонымен , Linux және Hurd бір-біріне қарама-қарсы келе жатыр. Бұл Mach микроядро серверлері болады.

Linux OSF микроядро негізінде

APPLE фирмасы MkLinux - Linux серверінің OSF микроядросына жұмысын демеуші болды. Бұл жұмысты OSF Resarch Institute орындады. Барлық шығатын Intel және Power Macintosh платформаларына арналған тексттер еркін таралады. – Linux сервер GNU GPL лицензиясымен таралады, микроядро және басқа серверлер OSF Free Copyright лицензиясымен таралады. Power Macintosh платформасына арналған нұсқа 1996 жылдың жазында дайын болды.

7.2.4. Soft Kernel

Soft Kernel жүйесі Microprocess (Courbevoie, France) фирмасымен өндіріледі. Басты сипаттамалары:

1. Түрі: host/target
2. Архитектура: объекті- микроядролар негізінде
3. Стандарт: өзіндік
4. Қасиеттері НУОЖ-нікіндей:
 - көп міндеттілік
 - көп процессорлық
 - 255 деңгейлі приоритет
 - жоспарлау: приоритетті, FIFO
 - preemptive ядросы
5. ОЖ жасау (host): UNIX/Windows.
6. Процессорлар (target): Motorola 68xxx, Power PC , Intel 80960, ARM, SPARC.
7. Байланыс сызықтары host/target: тізбекті канал, Ethernet
8. Ең аз өлшемі: 40 Kb
9. Синхрондау және өзара әсерлесу құралдары: семафорлар, белгілер, оқиғалар, пошталық жәшіктер.
10. Жасау құралдары:
 - SoftWorks - C/C++ жасаудың интегралданған ортасы

СДЖ арналған бақылау жұмыстары:

1. Қандай НУОЖ классикалық ОЖ-ге, қандай НУОЖ объекті—бейімделгенге жатады?
2. QNX жүйесінің басты сипаттамалары
3. Нақты уақыт қандай компоненттері QNX ОЖ-н қамтамасыз етеді?
4. Қандай екі іргелі принципке QNX ОЖ негізделген?
5. QNX ОЖ-нің микросының басты функциялары
6. QNX ОЖ конфигурациясын қандай менеджерлер қолданады?
7. Құрылғылар драйверлері деген не және олар қандай QNX ОЖ-де қызметтер атқарады?
8. pSOS ОЖ-нің басты сипаттамалары

9. VxWorks ОЖ-нің басты сипаттамалары
10. OS-9 ОЖ-нің басты сипаттамалары
11. OS-9 ОЖ қандай программалау тілінде жазылған?
12. OS-9 ОЖ жасау кезінде қандай талаптар қойылады?
13. OS-9 ОЖ – дегі модульділік деген не?
14. OS-9 ОЖ-ның ядросы қандай қызметтер атқарады?
15. OS-9 ОЖ-ң жүйе сенімділігін қанағаттандырады?
16. OS-9 ОЖ –де үзілулерді өңдеу программасы қалай жұмыс істейді?
17. Linux жүйесінің басты даму бағыттары
18. НУОЖ-ның даму тенденциялары

Ұсынылатын әдебиеттер:

1. А.В.Гордеев, А.Ю.Молчанов. Системное программное обеспечение. — "Питер", 2002. — 736с. Соломейчук В.Г. Linux. Экспресс курс. – СПб.: БХВ – Петербург, 2005. – 277с.
2. Стахнов А.А. Linux. Научное издание. 2-е изд. - СПб.: БХВ – Петербург, 2005. – 944с.
3. Зыль С.Н. Операционная система реального времени QNX: от теории к практике. 2-е изд., перераб. доп. – СПб.: БХВ – Петербург, 2004. – 191с.
4. Операционная система реального времени QNX Neutrino 6.3. Системная архитектура. пер. с англ./ гл. ред. Е. Кондукова. – М.: СПб.: Нижний Новгород: Питер, 2006. – 336с.
5. Зыль С.Н. QNX momentics. Основы применения. – СПб.: БХВ – Петербург, 2005. – 255с.
6. Болл Б., Хойт Д. Red Hat Linux.8/9. Настольная книга пользователя. пер. с англ. – М.: СПб.: Киев: DiaSoft, 2005. – 921с.
7. Цилюрик О., Горшко Е. QNX/UNIX. Анатомия параллелизма. – СПб.: М.: Символ, 2006. – 287с.

**4. Зертханалық жұмыстарды өткізуге арналған әдістемелік нұсқаулар
№1 зертханалық жұмысқа арналған әдістемелік нұсқаулар**

Жұмыстың мақсаты:

- технологиялық кестелерді, жұмыс нұсқауларын және көтермен орнатуды басқару процесін талдау;
- Көтермені орнатудың программалық имитаторын жасау

Зертханалық жұмыс 4 сағатқа белгіленген және «ЛИФТ» жасалаған компонентінің негізінде көтерме орнатудың программалық имитаторын құру бойынша практикалық тәжірибе алу және теориялық материалды оқу зертханалық жұмысты тапсыру, бақылау сұрақтарына жауап және дербес компьютерде жеке тапсырмаларды көрсетуден тұрады.

Отчеттің мазмұны:

1. Жұмыстың аты мен мақсаты
2. Жеке тапсырма
3. Бақылау сұрақтарына жауаптар

Жұмыстың орындалу тәртібі:

1. Технологиялық схеманы, жұмыстың нұсқасын және көтерме орнатуды басқару процесін талдау.
2. «ЛИФТ» компоненттің жұмысын зерттеу
3. Көтерме орнатудың басқару имитаторының диспетчерлік пункттен алгоритмдік программасын жасау

4. Көтерме орнатудың басқару имитаторының диспетчерлік пунктен код программасын жасау
5. Программаны жөндеуден өткізу

Бақылау сұрақтары:

1. ЛИФТінің басқару жүйесінің блок-схемасы қандай басты блоктардан тұрады? Әрбір блоктың міндетін суреттеңіз.
2. ЛИФТ жұмысының алгоритмін сипаттаңыз
3. Өзіндік графикалық компоненттерді құру процесін сипаттаңыз.
4. Қандай оператордың көмегімен әдістер, процедуралар енгізу/ шығару компонент қасиеттері орындалады?

Ұсынылатын әдебиеттер:

1. А.В.Гордеев, А.Ю.Молчанов. Системное программное обеспечение. — "Питер", 2002. — 736с.
2. Кристиан К. Введение в операционную систему Unix: пер. с англ. — М. Финансы и статистика, 1985. — 360с.
3. Робачевский А.М., Немнюгин С.А., Стесик О.Л. Операционная система Unix. 2-е изд.— СПб.: БХВ – Петербург, 2005. — 635с.

СДЖ-ге арналған бақылау тапсырмалары:

1. «ЛИФТ» компонентінің программалық кодын зерттеу
2. Меншікті әр түрлі мақсатта компоненттерді құрудағы жұмыс тәжірбиесін көрсетіңіз.

№2 зертханалық жұмысқа арналған әдістемелік нұсқаулар.

Технологиялық процестің имитаторын жасау

Жұмыстың мақсаты:

- технологиялық кестелерді, жұмыс принциптерін және технологиялық процестерді басқару процесін талдау;
- Технологиялық процестің программалық имитаторын жасау

Зертханалық жұмыс 4 сағатқа белгіленген және «ЛИФТ» жасалаған компонентінің негізінде көтерме орнатудың программалық имитаторын құру бойынша практикалық тәжірбие алу және теориялық материалды оқу зертханалық жұмысты тапсыру, бақылау сұрақтарына жауап және дербес компьютерде жеке тапсырмаларды көрсетуден тұрады.

Отчеттің мазмұны:

1. Жұмыстың аты мен мақсаты
2. Жеке тапсырма
3. Бақылау сұрақтарына жауаптар

Жұмыстың орындалу тәртібі:

1. технологиялық кестелерді, жұмыс принциптерін және технологиялық процестерді басқару процесін талдау;
2. технологиялық процессордың графикалық имитаторын жасау
3. технологиялық процессорды имитатормен басқару үшін диспетчерлік пунктен алгоритм және программа кодын жасау;
 - енгізілетін программалық процессордың көлемін беру үшін программалық коды бар панельді жасау;
 - қадағаланатын программалық процессордың көлемін беру үшін программалық коды бар қадағалайтын панельді жасау;
 - диспетчерлік пунктен процессорды басқару панельін жасау;
4. болу мүмкін апатты жағдайларды модульдеу, апатты панельді құру және ол үшін программалық код жасау.

Бақылау сұрақтары:

1. Технологиялық процесті басқару жүйесінің блок-схемасы қандай басты блоктардан тұрады? Әрбір блоктың міндетін суреттеңіз.
2. Технологиялық процессордың жұмысының алгоритмін суреттеңіз.

Ұсынылатын әдебиеттер:

1. А.В.Гордеев, А.Ю.Молчанов. Системное программное обеспечение. — "Питер", 2002. — 736с.
2. Кристиан К. Введение в операционную систему Unix: пер. с англ. — М. Финансы и статистика, 1985. — 360с.
3. Робачевский А.М., Немнюгин С.А., Стесик О.Л. Операционная система Unix. 2-е изд.— СПб.: БХВ – Петербург, 2005. — 635с.

СДЖ-ге арналған бақылау тапсырмалары:

1. Технологиялық процес үшін графикалық компоненті бар программалық кодты жасау

5. Студенттің оқытушымен дербес жұмысының тақырыптамалық жоспары

СОДЖ тақырыбының атауы	Сабақтың мақсаты	Сабақты өткізу түрі	Сабақтың мазмұны	Ұсынылатын әдебиет
1 Операциялық жүйелер және орталар. Жалпы түсініктер ОЖ-ның эволюциясы. ОЖ-ның негізгі функциялары	Берілген тақырып бойынша тереңдетілген білім	Сұрау	Тестілеу емтихан тестері бойынша өткізіледі және студенттің емтиханға дайындығына қосымша болып табылады	1, 2, 8-11
2 Ож-мен ұсынылатын сервистер. ОЖ-ның классификациясы. Ож-ны жобалаудың заманға сай концепциялары мен технологиялары	Берілген тақырып бойынша тереңдетілген білім	Сұрау		1, 2, 8-11
3 Қорлар түсінігі. Қорлардың классификациясы. Қорлардың негізгі түрлері	Берілген тақырып бойынша тереңдетілген білім	Сұрау, тестілеу		1, 2, 8-11
4 Есептеуіш процес ұғымы. Виртуальдылықтың концепциясы. Процестің күйінің диаграммасы. Тығырық пен критикалық секцияларды іске асыру.	Берілген тақырып бойынша тереңдетілген білім	Сұрау		1, 2, 8-11
5 процестер мен тредтер. Үзілулер.	Берілген тақырып	Сұрау, тестілеу		1, 2, 8-11

СОДЖ тақырыбының атауы	Сабақтың мақсаты	Сабақты өткізу түрі	Сабақтың мазмұны	Ұсынылатын әдебиет
Енгізу/шығаруды басқару.	бойынша тереңдетілген білім			
6 Операциялық жүйелерде міндеттерді басқару. Міндеттер мен процестерді жоспарлау және диспетчерлеу.	Берілген тақырып бойынша тереңдетілген білім	Сұрау		1, 2, 8-11
7 ОЖ-нің жадысын басқару. Жадының жіктелу әдісі.	Берілген тақырып бойынша тереңдетілген білім	Сұрау		1, 2, 8-11

6 Межелік бақылау және қорытынды аттестация кезінде студенттердің білімдерін бақылауға арналған материалдар

6.1 Пән бойынша жазба жұмысының тақырыбы

Жазба жұмыстары мен рефераттың тақырыбы

1. Нақты уақыт ОЖ мен жалпы қолданыстағы ОЖ салыстыру. Олардың ұқсастықтары мен айырмашылықтары.
2. НУОЖ-нің механизмдері мен құрылу ерекшеліктері.
3. Нақты уақыттың классикалық ОЖ
4. Нақты уақыттың объекті- бейімделген ОЖ
5. Программалау жүйесінің функциялау принциптері
6. Программалау жүйесінің қосымша мүмкіндіктері
7. Нақты уақыт ОЖ мен жалпы қолданыстағы ОЖнің негізгі функциялары
8. Қор мен есептеуіш процесс ұғымы
9. ОЖ-де енгізу/шығаруды басқару
10. Тығырық пен критикалық секцияларды іске асыру
11. Процестер мен тредтер. Үзілулер
12. ОЖ-де міндеттерді басқару
13. ОЖ-де жадыны басқару

6.2 Өз-өзін тексеруге арналған сұрақтар (тестік тапсырмалар)

1. Терминалдар дегеніміз не? Мысал келтір
2. Нақты уақытта жұмыс атқаратын жүйелер өзіндік байланысы бар жүйелер арасында қандай айырмашылық бар?
3. Нақты уақыт жүйесін қолданудың қандай 2 тобы (категориясы) бар?
4. Нақты уақыт жүйесінің мақсаты не болып табылады?
5. Нақты уақыт жүйесіне бірнеше анықтама бер. Олардың қайсысы нақтырақ? Не себепті?
6. Жұмсақ нақты уақыт жүйесі деген не?
7. Қатты нақты уақыт жүйесі деген не?
8. Қатты және жұмсақ нақты уақыт жүйелері арасында қандай айырмашылықтар мен ұқсастықтар бар?

9. Self-Hosted НУОЖ. Оның артықшылықтары мен кемшіліктері.
10. Host/Target НУОЖ. Оның артықшылықтары мен кемшіліктері.
11. Шығу тегіне байланысты НУОЖ қандай түрлері болады? Олар қандай түрге жатады және неліктен?
12. Жүйелік бағдарламалық ортаға байланысты НУОЖ қандай кластарға бөлінеді?
13. Ішкі құрылымы бойынша НУОЖ қандай болады?
14. НУОЖ мен жалпы қолданыстағы ОЖ негізгі айырмашылықтары
15. Нақты уақыттағы орындалу жүйесі деген не? (анықтама бер)
16. Нақты уақыттағы орындалу жүйесіне қандай талаптар қойылады?
17. Нақты уақыттағы жетілдіру жүйесі деген не? (анықтама бер)
18. Нақты уақыттағы жетілдіру жүйесі неден құралады?
19. НУОЖ өзегі қандай қызметтер атқарады?
20. Қорды синхрондау қалай жүзеге асырылады?
21. Міндетаралық айырбас қалай жүзеге асырылады?
22. Мәліметтердің жіктелінуі деген не?
23. Сыртқы құрылғылардың сұраныстарын өңдеу қалай жүзеге асырылады?
24. Ерекше жағдайларды өңдеу не үшін қажет?
25. НУОЖ-дегі қай жоспарлау принципі жиі қолданылады?
26. Жоспарлау кезінде қандай міндеттер жиі атқарылады?
27. Міндеттердің өзара үйлесімділігі қалай атқарылады?
28. Қорларды пайдалану кезінде қандай қиындықтар туады? Критикалық секция деген не?
29. Сыртқы оқиғаларды синхрондау кезінде қандай негізгі беталыстар қолданылады?
30. Уақыт бойынша синхрондау не үшін қолданылады?
31. Процестер мен мәліметтер тасымалын синхрондау құралдарына не жатады?
32. Жүйе реакциясының оқиғаға қатысты уақыты болып не табылады?
33. Жүйе реакциясының уақыты кезінде қандай уақыт аралығы аппаратурамен анықталады?
34. Жүйе реакциясының уақыты кезінде қандай уақыт аралығы операциялық жүйе мен компьютер архитектурасымен анықталады?
35. UNIX ОЖ-нің басты артықшылықтары қандай?
36. UNIX ОЖ қандай 2 үлкен бөліктерге бөлінеді?
37. UNIX ОЖ-нің тұтынушылары
38. UNIX ОЖ-нің тұтынушы интерфейсі
39. UNIX ОЖ-нің артықшыланған тұтынушысы
40. UNIX ОЖ –нің программалары
41. UNIX ОЖ-нің командалары
42. UNIX ОЖ-нің процестері
43. UNIX ОЖ-нің енгізу / шығару бағытының өзгеруі
44. UNIX ОЖ-нің аталған программалық каналдары
45. UNIX ОЖ-нің қорғаныс принциптері
46. UNIX ОЖ-нің құрылғыларын басқару
47. UNIX ОЖ-нің құрылғылар драйверлері
48. UNIX ОЖ-нің құрылғыларының ішкі және сыртқы интерфейстері
49. UNIX ОЖ-нің программалық ұялары (Sockets)
50. UNIX ОЖ-нің жойылған процедураларды (RPC) шақыруы
51. UNIX ОЖ қандай компоненттерден тұрады?
52. UNIX ОЖ-нің басты функциялары
53. UNIX ОЖ-ің shell басты функциялары
54. UNIX ОЖ-нің командаларының басты функциялары
55. UNIX ОЖ-нің файлдық жүйесінің басты функциялары

56. UNIX ОЖ-нің мобильділігіне қалай қол жеткізуге болады?
57. UNIX ОЖ-нің өзегінің машиналы-тәуелді бөлігіне нені жатқызуға болады?
58. UNIX ОЖ-нің өзегінің басты функциялары қандай?
59. Тұтынушы режимі мен UNIX ОЖ-нің өзегінің режимінің айырмашылықтары неде?
60. Өзекпен өзара әсерлесу принциптері. Жүйелік шақырулар деген не?
61. UNIX ОЖ-де файлдың қандай түрлері бар?
62. Кәдімгі файлдар мен анықтамалар арасында қандай айырмашылық бар?
63. Арнайы файлдар не үшін қолданылады? Арнайы файлдың қандай түрлері бар? Олар қайда қолданылады?
64. UNIX ОЖ-де бірдей файлдар әр түрлі атпен байланыса ала ма?
65. UNIX ОЖ-де виртуальды жадыда файлдарды сипаттау қалай жүргізіледі?
66. UNIX ОЖ-ның бұрынғы нұқаларында файлдарға рұқсатты синхрондау қалай жүргізілген? Қазіргі кезде синхрондаудың қандай нұсқалары қолданылады?
67. UNIX ОЖ-де рұқсат етілмеген қол жеткізуден файлдарды қорғау неге негізделген?
68. Жіктелген файлдық жүйе деген не? Оның кемшіліктері мен артықшылықтары
69. Ағынды драйвердің жұмыс істеу принципі
70. TCP/IP хаттамалар стегі
71. UNIX ОЖ-де виртуальді жадының қандай ұйымдастыру әдістері қолданылады? Оларға сипаттама бер
72. UNIX ОЖ-де виртуальді жадының қандай секіріс (подкачка) қандай ұйымдастыру әдістері қолданылады?
73. Әрбір процестің виртуальды жадысы қандай сегменттермен берілген? Оларға қысқаша сипаттама бер
74. Аппаратты – тәуелсіз деңгейде сегметті жадының ұйымдастыру әдістері қалай жүзеге асырылады?
75. UNIX ОЖ-де басты жадының беттік орнын басу қалай жүзеге асырылады?
76. UNIX ОЖ -де процестердің тууы қалай іске асырылады?
77. UNIX ОЖ -де басымдықтың неше және қандай түрлері бар?
78. UNIX ОЖ -де құрастырмалы блоктардың қандай элементтері қолданылады?
79. Тұтынушылық және ядролық құрамды процестер
80. UNIX ОЖ -де процестерді қандай жоспарлау алгоритмі қолданылады?
81. Тұтынушы деңгейінде процестерді басқару механизмінің дәстүрлі түрі бар?
82. UNIX ОЖ -дегі тізбек ұғымы
83. UNIX ОЖ -де енгізу/шығару ұйымының қандай басты түрлері бар?
84. UNIX ОЖ -дегі семафорлар
85. UNIX ОЖ -дегі хаттар кезегі
86. UNIX ОЖ -дегі үзілулер мен ерекше жағдайлар
87. UNIX ОЖ-нің ортасын қабылдайтын келешек ОЖ
88. Windows NT – ның басты сипаттамалары
89. Windows NT-да микроядро концепциясы не арқылы жүзеге асырылады?
90. Windows NT-да қандай 2 режим бар?
91. Windows NT-да сервистер мен серверлер қандай режимдерде орындалады және олардың өзара әсерлесулері қалай орындалады?
92. Серверлердің, сервистердің, енгізу/ шығару менеджерлерінің , микроядро және аппаратты абстракция деңгейлерінің негізгі қызметтері.
93. Windows NT-ның барлық программаларын біртұтас жүйеге біріктіретін басты құрал не?
94. ОЖ-нің қандай қолданбалы орталдарын Windows NT қабылдайды?
95. Объектіленер деген не, олар не үшін қолданылады және олар қандай бөліктерден тұрады?

96. Windows NT-да процестер мен тізбектер қалай жүзеге асырылады?
97. Windows NT-дағы тізбектердің жоспарлау алгоритмдері
98. Windows NT-нің интерактивті есептер үшін қандай жоспарлау жүзеге асырылады?
99. Windows NT-дағы үзілулерді өңдеу
100. Windows NT-дағы желілік құралдар
101. NT НУОЖ ретінде қолдану мүмкіндігі. «Иә» және «жоқ» аргументтері
102. NT НУОЖ ретінде қолдану мүмкіндігінің 2 шешімі
103. Қандай НУОЖ классикалық ОЖ-ге, қандай НУОЖ объекті—бейімделгенге жатады?
104. QNX жүйесінің басты сипаттамалары
105. Нақты уақыт қандай компоненттері QNX ОЖ-н қамтамасыз етеді?
106. Қандай екі іргелі принципке QNX ОЖ негізделген?
107. QNX ОЖ-нің микросының басты функциялары
108. QNX ОЖ конфигурациясын қандай менеджерлер қолданады?
109. Құрылғылар драйверлері деген не және олар қандай QNX ОЖ-де қызметтер атқарады?
110. рSOS ОЖ-нің басты сипаттамалары
111. VxWorks ОЖ-нің басты сипаттамалары
112. OS-9 ОЖ-нің басты сипаттамалары
113. OS-9 ОЖ қандай программалау тілінде жазылған?
114. OS-9 ОЖ жасау кезінде қандай талаптар қойылады?
115. OS-9 ОЖ – дегі модульділік деген не?
116. OS-9 ОЖ-ның ядросы қандай қызметтер атқарады?
117. OS-9 ОЖ-ң жүйе сенімділігін қанағаттандырады?
118. OS-9 ОЖ –де үзілулерді өңдеу программасы қалай жұмыс істейді?
119. Linux жүйесінің басты даму бағыттары
120. НУОЖ-ның даму тенденциялары

Модульдің №	Дәріс тақырыбының аты	Сұрақтың нөмері	СОДЖ тақырыбының аты	Сұрақтың нөмері
1	НУОЖ анықтамалары мен негізгі ұғымдары. НУОЖ мен жалпы қолданыстағы ОЖ-ның айырмашылығы. НУОЖ –ның құрылу ерекшеліктері. Нақты уақыт механизмдері	1-26	Операциялық жүйелер және орталар. Жалпы түсініктер ОЖ-ның эволюциясы. ОЖ-ның классификациясы	1-43
2	UNIX ОЖ	27-78	Есептеуіш процесс пен қор ұғымы. ОЖ-да енгізу/шығаруды басқару.	44-119
3	Windows NT. "Классикалық" және объекті-бейімделген НУОЖ.	79-109	ОЖ-да жады мен міндеттерді басқару.	120-155
	Барлығы:	109	Барлығы:	155

6.3 Дәрістің материалы бойынша тесттер

1. Терминалдар дегеніміз –ЭЕМ –ге мәліметтерді жіберіп және өңделген ақпаратты қабылдайтын құрылғылар деген анықтама дұрыс па?
 1. Иә
 2. Жоқ
2. Енгізілетін мәліметтер өздерінің пайда болатын нүктесі және/немесе тікелей ЭЕМ-ге енгізілетін жүйе, мұнда қолданылатын ортасына тікелей берілетін мәліметтер қалай аталады?
 1. Өзіндік байланысты жүйе
 2. Нақты уақытта жұмыс істейтін жүйелер
 3. Жалпы қолданыстағы жүйе
 4. Кросс-жетілдіру жүйесі
 5. Тұрақты дайындықты жүйе
3. Ақпаратты ала отырып, сыртқы объектілерді басқаратын, оны өңдеп кепілденген және қарастырылған қосымша үшін жеткілікті уақыт ішінде сыртқы объектінің функциясына әсер ету үшін нәтижесін қайтаратын жүйе дегеніміз-
 1. Нақты уақытта жұмыс істейтін жүйелер
 2. Өзіндік байланысты жүйе
 3. Жалпы қолданыстағы жүйе
 4. Кросс-жетілдіру жүйесі
 5. Тұрақты дайындықты жүйе
4. Анықтамалардың қайсысы НУОЖ үшін нақтығырақ:
 1. НУОЖ- сыртқы оқиғалардың болжамсыз ағынына болжамды уақыт аралығында төтеп беретін аппаратты – бағдарламалы кешен.
 2. НУОЖ- тұрақты дайын жүйе немесе жеткілікті реакция уақыты бар интерактивті жүйелер
 3. НУОЖ- енгізілетін мәліметтер өздерінің пайда болатын нүктесі тікелей ЭЕМ-ге енгізілетін жүйе, мұнда қолданылатын ортасына тікелей берілетін мәліметтер
 4. НУОЖ- компьютер қорларының тұтынушы мен орындалушы процестер арасындағы қолайлы жіктелуге бағытталған
 5. НУОЖ- объектде болатын аппаратурамен, объектімен, оқиғамен әрқашан байланысты жүйелер
5. Кез –келген «жылдам жүйе» бұл «нақты уақыт жүйесі» деген пікір дұрыс па?
 1. Иә
 2. Жоқ
6. Қатты нақты уақыт жүйесі бұл:
 1. Оқиғаға реакциямен ешқашан кешікпейтін жүйелер
 2. Оқиғамен реакция кезінде ешқана кешікпеуге тиіс жүйелер
 3. Жүйенің өзінде жұмыс істей отырып, тұтынушылар қосымшаны жасай алатын жүйелер
 4. нақты уақыт міндеттерін қолдануға арналған арнайы операциялық жүйелер
 5. Оның өндірушісімен жасалаған нақты микроконтроллер жүйесі
7. Жұмсақ нақты уақыт жүйесі бұл:
 1. Оқиғамен реакция кезінде ешқана кешікпеуге тиіс жүйелер
 2. Оқиғаға реакциямен ешқашан кешікпейтін жүйелер
 3. Жүйенің өзінде жұмыс істей отырып, тұтынушылар қосымшаны жасай алатын жүйелер

4. нақты уақыт міндеттерін қолдануға арналған арнайы операциялық жүйелер
 5. Оның өндірушісімен жасалаған нақты микроконтроллер жүйесі
8. Self – hosted НУОЖ
1. Файлдық жүйені қолдау тиіс, енгізу – шығару құрылғылары, қолданушы интерфейсі, компиляторы болуы жөнделуші прогамманы талдау құрылғылары, тексттік редактор
 2. Нақты уақыт қосымшасы host – компьютерде жасалады, кейін компоненттеледі және орындау үшін бүтін жүйеге жүктелетін жүйелер
 3. нақты уақыт міндеттерін қолдануға арналған арнайы операциялық жүйелер
 4. Оның өндірушісімен жасалаған нақты микроконтроллер жүйесі
 5. Объекті – бейімделген тілде жазылған, осы тілдер үшін кітапханалар жиыны және өңдеу құрылғыларын ұсынатын жүйелер
9. Host/Target НУОЖ
1. Нақты уақыт қосымшасы host – компьютерде жасалады, кейін компоненттеледі және орындау үшін бүтін жүйеге жүктелетін жүйелер
 2. Файлдық жүйені қолдау тиіс, енгізу – шығару құрылғылары, қолданушы интерфейсі, компиляторы болуы жөнделуші прогамманы талдау құрылғылары, тексттік редактор
 3. нақты уақыт міндеттерін қолдануға арналған арнайы операциялық жүйелер
 4. Оның өндірушісімен жасалаған нақты микроконтроллер жүйесі
 5. Объекті – бейімделген тілде жазылған, осы тілдер үшін кітапханалар жиыны және өңдеу құрылғыларын ұсынатын жүйелер
10. Объекті-бейімделген НУОЖ-
1. Объекті – бейімделген тілде жазылған, осы тілдер үшін кітапханалар жиыны және өңдеу құрылғыларын ұсынатын жүйелер
 2. Нақты уақыт қосымшасы host – компьютерде жасалады, кейін компоненттеледі және орындау үшін бүтін жүйеге жүктелетін жүйелер
 3. нақты уақыт міндеттерін қолдануға арналған арнайы операциялық жүйелер
 4. Оның өндірушісімен жасалаған нақты микроконтроллер жүйесі
 5. Файлдық жүйені қолдау тиіс, енгізу – шығару құрылғылары, қолданушы интерфейсі, компиляторы болуы жөнделуші прогамманы талдау құрылғылары, тексттік редактор

11. НУОЖ-ның 4 класы мен оның сипаты арасындағы сәйкестікті табыңыз.

Микропроцессор деңгейінде программалау	Әр түрлі құрылғыларда орнатылатын микропроцессорларды программалауға арналған программалар, өте үлкен және әдетте ассамблет типінің төменгі деңгей тілінде жазылған
Нақты уақыт жүйесінің минимальді ядросы	Негізгі функциялар ғана қарастырылған, ал жадыны басқару және диспетчер көбінесе қол жетімсіз. Қолданбалы программа құрал – жабдық ортасында жасалады, ереже бойынша орнатылған жүйелерде жасалады.
Нақты уақыт жүйесінің ядросы және құрал-жабдық ортасы	Жасау құрал-жабдық ортасында орындалады, ал орындалу бүтін жүйелерде орындалады. Көбінесе программаны параллельді орындауға қол жеткізіледі.

Толық сервері бар ОЖ	Сол жүйе рамкасында қолданбалы программаның орындалуы мен жасалуы жүргізіледі.
----------------------	--

12. Анықтамалардың қайсысы НУОЖ-ге қатысты-

1. Объектіде болып жатқан объект және оқиға аппаратурамен байланысты
2. Сол не басқа аппаратты – программалық кешен үшін тек құрал – жабдық ретінде қызмет етеді
3. Өзіне датчиктерді, енгізу – шығару моделдерін және компьютерін қосады.
4. Тұтынушы мен орындалатын процестер арасындағы компьютер қорын бөлісуге оптимальды түрде бейімделген
5. Дайын қосымшалар жиынтығы

13. ОЖ – ң жалпы міндетінің анықтамасына қайсысы қатсты:

1. Тұтынушы мен орындалатын процестер арасындағы компьютер қорын бөлісуге оптимальды түрде бейімделген
2. Дайын қосымшалар жиынтығы
3. Объектіде болып жатқан объект және оқиға аппаратурамен байланысты
4. Өзіне датчиктерді, енгізу – шығару моделдерін және компьютерін қосады.
5. Сол не басқа аппаратты – программалық кешен үшін тек құрал – жабдық ретінде қызмет етеді

14. Нақты уақыт операциялық жүйелерде орындалатын жүйе бұл:

1. Нақты уақыт қосымшасын функциялауды қамтамасыз ететін құрал – жабдықтар жиыны
2. Нақты уақыт қосымшасын жөндеу мен құрауды қамтамасыз ететін құралдар жиыны
3. Нақты уақыт қосымшасын басқару жүйесіндегі жұмыс ағынын реттейтін операциялық жүйенің бір бөлігі болатын программа
4. Нақты уақыт қосымшасының қорының жіктелуін координациялайтын басқарушы программа бөлігі
5. Нақты уақыттың басқа қосымшаларын орындалуын басқаратын және осы қосымшалардың ағымын реттейтін басқарушы программа бөлігі

15. Нақты уақыт операциялық жүйесін жасайтын жүйе бұл :

1. Нақты уақыт қосымшасын жөндеу мен құрауды қамтамасыз ететін құралдар жиыны
2. Нақты уақыт қосымшасын функциялауды қамтамасыз ететін құрал – жабдықтар жиыны
3. Нақты уақыт қосымшасын басқару жүйесіндегі жұмыс ағынын реттейтін операциялық жүйенің бір бөлігі болатын программа
4. Нақты уақыт қосымшасының қорының жіктелуін координациялайтын басқарушы программа бөлігі
5. Нақты уақыттың басқа қосымшаларын орындалуын басқаратын және осы қосымшалардың ағымын реттейтін басқарушы программа бөлігі

16. Нақты уақыт жүйесін орындайтын ортаға қойылатын талаптар, бұл:

1. Жүйенің шағын жадысы, жадыдағы жүйенің резиденттілігі, көпміндеттілік, үзілуге қызмет ететін приоритеті бар ядро, приоритеті бар диспетчер
2. Шағын жады, дичпетчерлеу- кооперативті көпміндеттілік, толықтыруды қолдану және беттерді жадыда орнату, көп міндеттілік, үзілуге қызмет ететін приоритеті бар ядро

3. Жүйенің шағын жадысы, жадыдағы жүйенің резиденттілігі, операцияның аяқтаудың өзіндік орташа өлшем, үзілуге қызмет ететін приоритеті бар ядро, диспетчерлеу- кооперативті көпміндеттілік
4. Жүйенің шағын жадысы, толықтыруды қолдану , үзілуге қызмет ететін приоритеті бар ядро, приоритеті бар диспетчер
5. Жүйенің шағын жадысы, жадыдағы жүйенің резиденттілігі, көпміндеттілік, үзілуге қызмет ететін приоритеті бар ядро, «карусель» типті диспетчерлеу

17. Нақты уақыт жүйесінің орындалу жүйесі неден тұрады?

1. Ядро, драйверлер, орындалушы модульдер
2. Жойылған жөндеу құралдары, профильдену , модельдеу, бүтін процессор эмуляциясы, өзара әсерлесуші міндеттерді жөндеу арнайы құралдары
3. Жоғары деңгейлі құралдар, файлдық жүйе, желілік қолдау, драйверлер, орындалушы модульдер
4. Ядро, драйверлер, жоғары деңгейлі құралдар, бүтін процессор эмуляциясы, орындалушы модульдер
5. Жоғары деңгейлі құралдар, профильдену, бүтін процессор эмуляциясы, драйверлер, орындалушы модульдер

18. Нақты уақыт операциялық жүйесін жасайтын жүйе неден тұрады?

1. Жойылған жөндеу құралдары, профильдену , модельдеу, бүтін процессор эмуляциясы, өзара әсерлесуші міндеттерді жөндеу арнайы құралдары
2. Ядро, драйверлер, орындалушы модульдер
3. Жоғары деңгейлі құралдар, файлдық жүйе, желілік қолдау, драйверлер, орындалушы модульдер
4. Ядро, драйверлер, жоғары деңгейлі құралдар, бүтін процессор эмуляциясы, орындалушы модульдер
5. Жоғары деңгейлі құралдар, профильдену, бүтін процессор эмуляциясы, драйверлер, орындалушы модульдер

19. НУОЖ ядросы келесі сервистерді қамтамасыз ете алады:

1. Қорларды синхрондау, міндетаралық айырбас, мәліметтерді бөлісу, сыртқы құрылғы сұранысын жасау, ерекше жағдайларды жасау
2. Қорларды синхрондау, міндетаралық айырбас, мәліметтерді бөлісу, сыртқы құрылғы сұранысын жасау, желілік қолдау, ерекше жағдайларды жасау
3. Қорларды синхрондау, мәліметтерді бөлісу, сыртқы құрылғы сұранысын жасау, ерекше жағдайларды жасау
4. Қорларды синхрондау, файлдық жүйе, мәліметтерді бөлісу, сыртқы құрылғы сұранысын жасау, ерекше жағдайларды жасау
5. Сыртқы жағдайларды синхрондау, міндетаралық айырбас, мәліметтерді бөлісу, сыртқы құрылғы сұранысын жасау, ерекше жағдайларды жасау

20. Ұғым мен оның сипаты арасындағы сәйкестікті табыңыз.

Қорларды синхрондау	Семафорға қол жеткізілгенге дейін процесс күту керек, керекті қор босағанда осы қорды алып семафорға орнату керек
Міндетаралық айырбас	Ішкі байланыс хат жіберу жүйесімен жүзеге асырылады. Сыртқы байланыс- байланыс желілерімен орындалады.
Мәліметтерді бөлісу	Жүйеде жалпы жады бөлімдеріне қол жетімді және кезектің әр түрлі типтерін ұйымдастыру қарастырылады

сыртқы құрылғы сұранысын жасау	ядро енгізу/шығару қызметін қамтамасыз етеді және драйверлік құрылғылармен жұмысты жеңілдететін серверді ұсынады.
ерекше жағдайларды жасау	Синхронды және асинхронды жағдайларды жасау мүмкіндігін ұсыну нақты уақыттың тез және дәл ішкі және сыртқы жағдайларға жауап беруге мүмкіндік береді.

21. НУОЖ – да көп қолданылатын жоспарлау принципі бұл:

1. Ығысырушы приоритетті көпміндеттілік
2. Циклдік алгоритм
3. Тең қол жетімділікпен уақыт бөлісу
4. Кооперативті көпміндеттілік
5. Ығыстылмайтын көпміндеттілік

22. Ұғым мен оның сипаты арасындағы сәйкестікті табыңыз.

Байланыс міндеттерін басқару қамтамасыз етеді	Хаттар арқылы міндеттердің өзара қиылысуы
Жалпы қорларды басқару қамтамасыз етеді	Бір уақытта физикалық құрылғыны қолдану немесе жадының тек бір міндетті облысын қолдану
Сыртқы оқиғалармен синхрондау қамтамасыз етеді	Сыртқы оқиғаға жүйенің максималды тез реакциясы, минимум уақыт, егер жүйеде үзілулер тиым салынған болса, онда үзілуді жасау қосымша программалары максималды аз уақыт ішінде минималды функция көлемін орындайды
Уақыт бойынша синхрондау қамтамасыз етеді	Міндеттің орындалуын белгілі бір уақытқа дейін тоқтату, уақыт бойынша бірнеше міндетті синхрондауға қатысты белгілі уақытта міндеттерді іске қосу

23. Мәліметтерді жіберу мен процесті синхрондау құрылғыларына жататындар:

1. Семафорлар, мьютекстер, оқиғалар, белгілер, бөлінетін мәліметтер жұмысы үшін құралдар, мәліметтер каналы
2. Ығыстырушы приоритетті көпміндеттілік, құрылғы драйвері, жоғары деңгей құралдары, профилдеу, орындалатын модульдер
3. Семафорлар, мьютекстер, оқиғалар, құрылғы драйвері, жоғары деңгей құралдары
4. оқиғалар, белгілер, бөлінетін мәліметтер жұмысы үшін құралдар, мәліметтер каналы, жоғары деңгей құралдары, профилдеу, орындалатын модульдер
5. мьютекстер, оқиғалар, белгілер, құрылғы драйвері, жоғары деңгей құралдары, профилдеу

24. Оқиғаға жүйенің уақытпен реакциясы не болып табылады?

1. Уақыт интервалы- объектідегі оқиғадан осы оқиғаның жасау программасын бірінші инструкциясын орындағанға дейін
2. Уақыт интервалы- объектідегі оқиғадан үзілу генерациясына дейін
3. Уақыт интервалы- үзілудегі сұраныстың пайда болуынан өңдеушінің бірінші инструкциясын орындағанға дейін
4. Уақыт интервалы- объектідегі оқиғадан үзілудегі сұраныстың пайда болуына дейін

5. Уақыт интервалы-үзілу генерациясынан өңдеушінің бірінші инструкциясын орындағанға дейін

25. Жүйенің уақыт реакциясы кезінде қандай уақыт интервалы аппаратурамен анықталады?

1. Уақыт интервалы- объектідегі оқиғадан үзілу генерациясына дейін
2. Уақыт интервалы- объектідегі оқиғадан осы оқиғаның жасау программасын бірінші инструкциясын орындағанға дейін
3. Уақыт интервалы- үзілудегі сұраныстың пайда болуынан өңдеушінің бірінші инструкциясын орындағанға дейін
4. Уақыт интервалы- объектідегі оқиғадан жүйелік модульдің жұмысының басталуына дейін
5. Уақыт интервалы-үзілу генерациясынан өңдеушінің бірінші инструкциясын орындағанға дейін

26. Жүйенің уақыт реакциясы кезінде қандай уақыт интервалы операциялық жүйе мен компьютер архитектурасымен анықталады?

1. Уақыт интервалы- үзілудегі сұраныстың пайда болуынан өңдеушінің бірінші инструкциясын орындағанға дейін
2. Уақыт интервалы- объектідегі оқиғадан осы оқиғаның жасау программасын бірінші инструкциясын орындағанға дейін
3. Уақыт интервалы- объектідегі оқиғадан үзілу генерациясына дейін
4. Уақыт интервалы- объектідегі оқиғадан үзілудегі сұраныстың пайда болуына дейін
5. Уақыт интервалы- жүйелік модульдің жұмысының басталуынан өңдеушінің бірінші инструкциясын орындағанға дейін

27. UNIX ОЖ үшін шындық деген не?

1. Жаңа тұтынушыларды тіркеуді әдетте жүйе администраторы орындайды
2. Әрбір тіркелген тұтынушыға файлдық жүйенің кейбір каталогы сәйкес келеді.
3. тұтынушы өзінің паролін қоюға/өзгертуге болады
4. Жүйеге енгенде тұтынушы өз үй каталогына және барлық каталогтарға, файлдық жүйеге рұқсат алады.
5. Тұтынушы өзінің атын өзгерте алады.

28. UNIX ОЖ үшін шындық деген не?

1. Жүйеге енгенде тұтынушы өз үй каталогына және барлық каталогтарға, файлдық жүйеге рұқсат алады.
2. артықшыланған тұтынушы жүйенің барлық қорларын қолдана алады
3. артықшыланған тұтынушының идентификаторының мәні ноль
4. артықшыланған тұтынушы тұтынушы тобының біріне жатады
5. артықшыланған тұтынушы жүйені толық бақылай алмайды

29. UNIX ОЖ үшін шындық деген не?

1. UNIX ОЖ қолданбалы программаларды орындаудың операциялық ортасы болып табылады
2. UNIX ОЖ жаңа қосымшаларды жасау ортасы болып табылады
3. UNIX ОЖ ортасында стандартты программалау тілі «C» тілі болып табылады
4. UNIX ОЖ қолданбалы программаларды орындаудың операциялық ортасы болып табылмайды
5. UNIX ОЖ жаңа қосымшаларды жасау ортасы болып табылмайды

30. UNIX ОЖ-де процесс дегеніміз-

1. виртуальды адресі кеңістікте орындалатын программа
2. әлдебір файлға енгізу және әлдебір файлға шығару
3. желіаралық өзара әсерлесулердің дәстүрлі құралдарының бірі
4. құрылғымен жұмысты иницирлей алатын өз статикалық деректері бар көп кірісті программалық модуль
5. кез келген программаға басқа программамен хат алысуға мүмкіндік беретін құрылғы

31. UNIX ОЖ-дегі құрылғылар драйвері

1. құрылғымен жұмысты иницирлей алатын, тұтынушы мен берілген айырбасты орындап, құрылғыдан болған үзілулерді өңдейтін, көп кірісті программалық модуль
2. туынды байттар тізбегін қабылдауға және жіберуге бейімделген көп кірісті программалық модуль
3. UNIX ОЖ-нің ядросының блокты айырбасының жүйелік буферлеу мүмкіндіктерін қолданумен жұмыс істейтін көп кірісті программалық модуль
4. конвейер модульдерінен тұратын, тұтынушының сұраныстарының көп сатылы өңдеуін қамтамасыз ететін, желілік құрылғыларға рұқсатты іске асыруға арналған көп кірісті программалық модуль
5. құрылғымен жұмысты иницирлей алатын, тұтынушы мен берілген айырбасты орындайтын көп кірісті программалық модуль

32. UNIX ОЖ-нің символдық драйвері бұл:

1. туынды байттар тізбегін қабылдауға және жіберуге бейімделген көп кірісті программалық модуль
2. құрылғымен жұмысты иницирлей алатын, тұтынушы мен берілген айырбасты орындап, құрылғыдан болған үзілулерді өңдейтін, көп кірісті программалық модуль
3. UNIX ОЖ-нің ядросының блокты айырбасының жүйелік буферлеу мүмкіндіктерін қолданумен жұмыс істейтін көп кірісті программалық модуль
4. конвейер модульдерінен тұратын, тұтынушының сұраныстарының көп сатылы өңдеуін қамтамасыз ететін, желілік құрылғыларға рұқсатты іске асыруға арналған көп кірісті программалық модуль
5. құрылғымен жұмысты иницирлей алатын, тұтынушы мен берілген айырбасты орындайтын көп кірісті программалық модуль

33. UNIX ОЖ-нің блоктық драйвері бұл:

1. UNIX ОЖ-нің ядросының блокты айырбасының жүйелік буферлеу мүмкіндіктерін қолданумен жұмыс істейтін көп кірісті программалық модуль
2. құрылғымен жұмысты иницирлей алатын, тұтынушы мен берілген айырбасты орындап, құрылғыдан болған үзілулерді өңдейтін, көп кірісті программалық модуль
3. туынды байттар тізбегін қабылдауға және жіберуге бейімделген көп кірісті программалық модуль
4. конвейер модульдерінен тұратын, тұтынушының сұраныстарының көп сатылы өңдеуін қамтамасыз ететін, желілік құрылғыларға рұқсатты іске асыруға арналған көп кірісті программалық модуль
5. құрылғымен жұмысты иницирлей алатын, тұтынушы мен берілген айырбасты орындайтын көп кірісті программалық модуль

34. UNIX ОЖ-нің ағынды драйвері бұл:

1. конвейер модульдерінен тұратын, тұтынушының сұраныстарының көп сатылы өңдеуін қамтамасыз ететін, желілік құрылғыларға рұқсатты іске асыруға арналған көп кірісті программалық модуль
2. құрылғымен жұмысты иницирлей алатын, тұтынушы мен берілген айырбасты орындап, құрылғыдан болған үзілулерді өңдейтін, көп кірісті программалық модуль
3. туынды байттар тізбегін қабылдауға және жіберуге бейімделген көп кірісті программалық модуль
4. конвейер модульдерінен тұратын, тұтынушының сұраныстарының көп сатылы өңдеуін қамтамасыз ететін, желілік құрылғыларға рұқсатты іске асыруға арналған көп кірісті программалық модуль
5. құрылғымен жұмысты иницирлей алатын, тұтынушы мен берілген айырбасты орындайтын көп кірісті программалық модуль

35. UNIX ОЖ қандай құраушы бөліктерден тұрады?

1. ядро, shell, командалар, файлдық жүйе
2. ядро, shell, программалар, файлдық жүйе
3. ядро, shell, жүйелік шақырулар, құрылғы драйверлері
4. ядро, shell, программалық модульдер, файлдық жүйе
5. ядро, программалар, файлдық жүйелер, программалық модульдер

36. UNIX ОЖ – ң ядросының негізгі функцияларына жатады:

1. Жүйені инализациялау және тізбектер мен процесстерді басқару
2. Файлдар мен жадыны басқару
3. Коммуникативті құралдар және программалық интерфейс
4. Қызмет етуші программалар,ядро және тұтынушы режимі арасындағы контекстті процесті ауыстыру
5. Тексттерді өңдеу және ақпаратты ұйымдастыру

37. UNIX ОЖ – ң жүйесінің инициализациясы

1. компьютер жадысына толық ядроның жүктелуін қамтамасыз етеді және ядроны іске қосады
2. Процессор мен басқа компьютердің қорларының процесінің іске қосу арасында бөлінуді қамтамасыз етеді.
3. Шексіз процессордың виртуальды жадысын компьютердің физикалық оперативті жадысына бейнелейді
4. Функцияның кітапханасына түрінде безендірілген жүйелік шақырулар механизмі негізінде тұтынушы процестер жағына ядро мүмкіндігіне рұқсатты қамтамасыз етеді.
5. Процестер арасында, сонымен қатар процестер мен сыртқы құрылғылар драйверлері арасындағы мәліметтер тасымалын қамтамасыз етеді.

38. UNIX ОЖ – ң тізбектері мен процестерін басқару:

1. Процессор мен басқа компьютердің қорларының процесінің іске қосу арасында бөлінуді қамтамасыз етеді.
2. Шексіз процессордың виртуальды жадысын компьютердің физикалық оперативті жадысына бейнелейді
3. Функцияның кітапханасына түрінде безендірілген жүйелік шақырулар механизмі негізінде тұтынушы процестер жағына ядро мүмкіндігіне рұқсатты қамтамасыз етеді.
4. Процестер арасында, сонымен қатар процестер мен сыртқы құрылғылар драйверлері арасындағы мәліметтер тасымалын қамтамасыз етеді.

5. Компьютер жадысына толық ядроның жүктелуін қамтамасыз етеді және ядроны іске қосады
39. UNIX ОЖ-ның жадысын басқару
1. Шексіз процессордың виртуальды жадысын компьютердің физикалық оперативті жадысына бейнелейді
 2. Процессор мен басқа компьютердің қорларының процесінің іске қосу арасында бөлінуді қамтамасыз етеді.
 3. Функцияның кітапханасына түрінде безендірілген жүйелік шақырулар механизмі негізінде тұтынушы процестер жағына ядро мүмкіндігіне рұқсатты қамтамасыз етеді.
 4. Процестер арасында, сонымен қатар процестер мен сыртқы құрылғылар драйверлері арасындағы мәліметтер тасымалын қамтамасыз етеді.
 5. Компьютер жадысына толық ядроның жүктелуін қамтамасыз етеді және ядроны іске қосады
40. UNIX ОЖ-нің файлдарын басқару
1. файлдық жүйенің абстракциясын жүзеге асырады
 2. Компьютер жадысына толық ядроның жүктелуін қамтамасыз етеді және ядроны іске қосады
 3. Процессор мен басқа компьютердің қорларының процесінің іске қосу арасында бөлінуді қамтамасыз етеді.
 4. Шексіз процессордың виртуальды жадысын компьютердің физикалық оперативті жадысына бейнелейді
 5. Функцияның кітапханасына түрінде безендірілген жүйелік шақырулар механизмі негізінде тұтынушы процестер жағына ядро мүмкіндігіне рұқсатты қамтамасыз етеді.
41. UNIX ОЖ-ның коммуникативті құралдары
1. Процестер арасында, сонымен қатар процестер мен сыртқы құрылғылар драйверлері арасындағы мәліметтер тасымалын қамтамасыз етеді.
 2. Компьютер жадысына толық ядроның жүктелуін қамтамасыз етеді және ядроны іске қосады
 3. Процессор мен басқа компьютердің қорларының процесінің іске қосу арасында бөлінуді қамтамасыз етеді.
 4. Шексіз процессордың виртуальды жадысын компьютердің физикалық оперативті жадысына бейнелейді
 5. Функцияның кітапханасына түрінде безендірілген жүйелік шақырулар механизмі негізінде тұтынушы процестер жағына ядро мүмкіндігіне рұқсатты қамтамасыз етеді.
42. UNIX ОЖ-ның программалық интерфейсі
1. Функцияның кітапханасына түрінде безендірілген жүйелік шақырулар механизмі негізінде тұтынушы процестер жағына ядро мүмкіндігіне рұқсатты қамтамасыз етеді.
 2. Процестер арасында, сонымен қатар процестер мен сыртқы құрылғылар драйверлері арасындағы мәліметтер тасымалын қамтамасыз етеді.
 3. Компьютер жадысына толық ядроның жүктелуін қамтамасыз етеді және ядроны іске қосады
 4. Процессор мен басқа компьютердің қорларының процесінің іске қосу арасында бөлінуді қамтамасыз етеді.

5. Шексіз процессордың виртуальды жадысын компьютердің физикалық оперативті жадысына бейнелейді

43. UNIX жүйесінде тұтынушы процестерін орындау келесі деңгейлер бойынша іске асырылады:

1. тұтынушы және ядро
2. команда және программалар
3. супер-тұтынушы және ядро
4. команда және ядро
5. программа және ядро

44. Тұтынушы программалардың UNIX ОЖ –ң ядро функциясына көңіл аудару үшін қолданылады

1. Олардың орындалу кезінде, процессордың ішкі үзілуі пайда болатын, оны ядро режиміне көшіретін процессордың «арнайы командалары»
2. Супер тұтынушының арнайы кітапханасынан шақырылатын жүйелік функциялар
3. Компьютер жадысына толық ядроның жүктелуін қамтамасыз ететін және ядроны іске қосатын «арнайы командалар»
4. Төменгі деңгейлі жүйенің инициализациясы мен айналымының жүйелік функциялары
5. Процессордың ішкі және сыртқы үзілулерін алғашқы өңдеу командалары

45. UNIX ОЖ – ғы shell

1. Бұл сізге операциялық жүйемен байланысуға мүмкіндік беретін программа. Ол деректер ағынын басқару және олардың шартты орындалуын қамтамасыз ететін қуатты программалау тілі болып табылады
2. Процестің шексіз виртуалды жадысын компьютердің физикалық оперативті жадысына сипаттайтын функция
3. Процессор мен басқа компьютердің қорларының процесінің іске қосу арасында бөлінуді қамтамасыз етеді.
4. Компьютер жадысына толық ядроның жүктелуін қамтамасыз етеді және ядроны іске қосады
5. Алдын – ала трансляциясыз компьютер арқылы орындала алатын программалар

46. UNIX ОЖ – гі командалар

1. Алдын – ала трансляциясыз компьютер арқылы орындала алатын программалар
2. Процестің шексіз виртуалды жадысын компьютердің физикалық оперативті жадысына сипаттайтын функция
3. Процессордың уақыт процесінің іске қосу арасында бөлінуді қамтамасыз етеді
4. Қорлардың процестерінің іске қосу арасында бөлінуді қамтамасыз етеді
5. Компьютер жадысына толық ядроның жүктелуін қамтамасыз етеді және ядроны іске қосады

47. UNIX ОЖ – гі файлдық жүйе

1. Ұйымның логикалық әдісін, ақпараттың қалпына келтірілуін және басқарылуын қамтамасыз етеді, иерархиялық құрылымы бар
2. Процестің шексіз виртуалды жадысын компьютердің физикалық оперативті жадысына сипаттайтын функция

3. Процессор мен басқа компьютердің қорларының процесінің іске қосу арасында бөлінуді қамтамасыз етеді.
4. Компьютер жадысына толық ядроның жүктелуін қамтамасыз етеді және ядроны іске қосады
5. Өзіндік виртуалды адресті кеңістікте орындалатын программа

48. UNIX ОЖ – гі файл болуы мүмкін

1. Кәдімгі файл
2. Анықтама
3. Арнайы файл
4. Бір каналды
5. Көпканалды

49. UNIX ОЖ – гі кәдімгі файл

1. Символдар жиынтығы болып табылады және кез – келген ақпаратты сақтау үшін қолданылады
2. Символдар жиынтығы болып табылады және ақпаратты сақтау үшін қолданылмайды
3. Тек файлдардан ғана құралатын супер- файл болып табылады
4. Файлдар мен анықтамалардан құралатын супер- файл болып табылады
5. Мәліметтерді сақтамайды, бірақ файлдық жүйенің файлдық атауына физикалық ішкі құрылғыларды сипаттау механизмін қамтамасыз етеді

50. UNIX ОЖ – гі анықтамалар

1. Файлдар мен анықтамалардан құралатын супер- файл болып табылады
2. Символдар жиынтығы болып табылады және кез – келген ақпаратты сақтау үшін қолданылады
3. Символдар жиынтығы болып табылады және ақпаратты сақтау үшін қолданылмайды
4. Мәліметтерді сақтамайды, бірақ файлдық жүйенің файлдық атауына физикалық ішкі құрылғыларды сипаттау механизмін қамтамасыз етеді
5. Тек файлдардан ғана құралатын супер- файл болып табылады

51. UNIX ОЖ – гі арнайы файлдар

1. Мәліметтерді сақтамайды, бірақ файлдық жүйенің файлдық атауына физикалық ішкі құрылғыларды сипаттау механизмін қамтамасыз етеді
2. Символдар жиынтығы болып табылады және кез – келген ақпаратты сақтау үшін қолданылады
3. Символдар жиынтығы болып табылады және ақпаратты сақтау үшін қолданылмайды
4. Тек файлдардан ғана құралатын супер- файл болып табылады
5. Файлдар мен анықтамалардан құралатын супер- файл болып табылады

52. UNIX ОЖ – гі файлдарға паралеллді қол жеткізу кезіндегі синхрондау

1. Тұтынушылар мен процестер үшін міндетті емес
2. Барлық тұтынушылар мен процестер үшін міндетті
3. Барлық тұтынушылар үшін міндетті
4. Тұтынушылардың барлық процестері үшін міндетті
5. UNIX ОЖ – де синхрондау әдістері болмайды

53. UNIX ОЖ – де файлдарды қорғау келесі әдіс арқылы жүзеге асырылады

1. Файлда шкала сақталады, оның ішінде файлмен тұтынушы- оның иесі, оның иесі де , қалған тұтынушылар да тобына кіретін тұтынушылар не істей алатыны белгіленген
2. Файлды тек оның иелері ғана қолдана алады
3. Файлдарды тек қана оның иелері мен осы топтың тұтынушылары ғана қолдана алады
4. Файлдың қолдануына ешқандай шек қойылмайды
5. Файлдың қолдануына шек тек қана супер – тұтынушы арқылы қойылады

54. UNIX ОЖ – гі жіктелген файлдық жүйе кезінде

1. Жойылған файлдық жүйенің тұтынушылары жойылған файлдармен локальды файлдарға қарағанда баяу жұмыс істейді
2. Жойылған файлдық жүйенің тұтынушылары жойылған файлдармен локальды файлдарға қарағанда жылдам жұмыс істейді
3. Жойылған файлдық жүйенің тұтынушылары жойылған файлдарға қол жеткізе алмайды
4. Бір ғана локальды компьютердің қорлары пайдаланылады
5. Берілген локальды компьютердің барлық тұтынушыларының қорлары пайдаланылады

55. UNIX ОЖ – ң файлдық жүйесі әр түрлі аттары бар бірдей файлдарды байланыстыру мүмкіндігін қамтамасыз ете ала ма?

1. Иә
2. Жоқ

56. UNIX ОЖ – гі Streams

1. жүйелік шақырулар мен қосымша программалардан, сонымен қатар желілік өзара әсерлесу ұйымына арналған ядро қорларынан тұратын жалпы қолданыстағы байланысқан құралдар жиынтығы
2. Мәліметтерді сақтамайды, бірақ файлдық жүйенің файлдық атауына физикалық ішкі құрылғыларды сипаттау механизмін қамтамасыз етеді
3. Процессор мен басқа компьютердің қорларының процесінің іске қосу арасында бөлінуді қамтамасыз етеді.
4. Бұл сізге операциялық жүйемен байланысуға мүмкіндік беретін программа.
5. Функцияның кітапханасына түрінде безендірілген жүйелік шақырулар механизмі негізінде тұтынушы процестер жағына ядро мүмкіндігіне рұқсатты қамтамасыз етеді.

57. UNIX ОЖ – гі виртуалды жадының толықтыру алгоритмінің қандай түрлері бар?

1. Локальды,глобальды
2. FIFO , LRU
3. Жұмыс жиынтығы негізінде
4. SJN, LRU
5. Қатыстық приоритетті

58. UNIX ОЖ –гі глобальды толықтыру алгоритмі

1. Операциялық жүйе орын толтыру қажеттілігі кезінде барлық беттер арасынан басты жадының бетін іздейді
2. Орын толтыруға арналған бет сол виртуалды жадыға жазылған басты жадының беттерінің арасынан ғана ізделінеді

3. Орын толтыруға виртуалды жадыда көп сақталатын бет таңдап алынады
4. Орын толтыру көп уақыт көңіл бөлінбеген бет үшін орындалуы тиіс.
5. Операциялық жүйе әрбір уақыт аралығында процессорға қол жеткізілуі үшін бәсекелестік рұқсат етілген басты жадыдағы барлық процестердің болуын қамтамасыз етуі тиіс

59. UNIX ОЖ –гі локальды толықтыру алгоритмі

1. Орын толтыруға арналған бет сол виртуалды жадыға жазылған басты жадының беттерінің арасынан ғана ізделінеді
2. Операциялық жүйе орын толтыру қажеттілігі кезінде барлық беттер арасынан басты жадының бетін іздейді
3. Орын толтыруға виртуалды жадыда көп сақталатын бет таңдап алынады
4. Орын толтыру көп уақыт көңіл бөлінбеген бет үшін орындалуы тиіс
5. Операциялық жүйе әрбір уақыт аралығында процессорға қол жеткізілуі үшін бәсекелестік рұқсат етілген басты жадыдағы барлық процестердің болуын қамтамасыз етуі тиіс

60. UNIX ОЖ –гі FIFO толықтыру алгоритмі

1. Орын толтыруға виртуалды жадыда көп сақталатын бет таңдап алынады
2. Орын толтыруға арналған бет сол виртуалды жадыға жазылған басты жадының беттерінің арасынан ғана ізделінеді
3. Операциялық жүйе орын толтыру қажеттілігі кезінде барлық беттер арасынан басты жадының бетін іздейді
4. Орын толтыру көп уақыт көңіл бөлінбеген бет үшін орындалуы тиіс
5. Операциялық жүйе әрбір уақыт аралығында процессорға қол жеткізілуі үшін бәсекелестік рұқсат етілген басты жадыдағы барлық процестердің болуын қамтамасыз етуі тиіс

61. UNIX ОЖ –гі LRU толықтыру алгоритмі

1. Орын толтыру көп уақыт көңіл бөлінбеген бет үшін орындалуы тиіс
2. Орын толтыруға виртуалды жадыда көп сақталатын бет таңдап алынады
3. Орын толтыруға арналған бет сол виртуалды жадыға жазылған басты жадының беттерінің арасынан ғана ізделінеді
4. Операциялық жүйе орын толтыру қажеттілігі кезінде барлық беттер арасынан басты жадының бетін іздейді
5. Операциялық жүйе әрбір уақыт аралығында процессорға қол жеткізілуі үшін бәсекелестік рұқсат етілген басты жадыдағы барлық процестердің болуын қамтамасыз етуі тиіс

62. UNIX ОЖ –гі жұмыс жиынтығы негізіндегі толықтыру алгоритмі

1. Операциялық жүйе әрбір уақыт аралығында процессорға қол жеткізілуі үшін бәсекелестік рұқсат етілген басты жадыдағы барлық процестердің болуын қамтамасыз етуі тиіс
2. Орын толтыру көп уақыт көңіл бөлінбеген бет үшін орындалуы тиіс
3. Орын толтыруға виртуалды жадыда көп сақталатын бет таңдап алынады
4. Орын толтыруға арналған бет сол виртуалды жадыға жазылған басты жадының беттерінің арасынан ғана ізделінеді
5. Операциялық жүйе орын толтыру қажеттілігі кезінде барлық беттер арасынан басты жадының бетін іздейді

63. UNIX ОЖ –гі әрбір процестің виртуалды жадысы қанша сегмент түрінде ұсынылуы мүмкін

1. 5
2. 6
3. 7
4. 4
5. 3

64. UNIX ОЖ –нің виртуалды жадысының программалық кодының сегменті

1. Тек қана командалардан тұрады және процестің орындалу барысында жаңара алмайды
2. Берілген процесте орындалатын программаның инициализацияланған және инициализацияланбаған статикалық айнымалылардан тұрады.
3. Бөлінетін жадының сегментінің виртуалды жадыға қосылуы кезінде пайда болады
4. Бөлінетін сегменттердің түрлерінен құралады және оның беттері қажеттілік кезінде файл орналасқан ішкі жады облысында өз орнына жылжиды
5. Программаның автоматты айнымалылары анық және анық емес орналасқан виртуалды жады облысы

65. UNIX ОЖ –нің виртуалды жадысының мәліметтер сегменті

1. Берілген процесте орындалатын программаның инициализацияланған және инициализацияланбаған статикалық айнымалылардан тұрады
2. Тек қана командалардан тұрады және процестің орындалу барысында жаңара алмайды
3. Бөлінетін жадының сегментінің виртуалды жадыға қосылуы кезінде пайда болады
4. Бөлінетін сегменттердің түрлерінен құралады және оның беттері қажеттілік кезінде файл орналасқан ішкі жады облысында өз орнына жылжиды
5. Программаның автоматты айнымалылары анық және анық емес орналасқан виртуалды жады облысы

66. UNIX ОЖ –нің виртуалды жадысының стек сегменті

1. Программаның автоматты айнымалылары анық және анық емес орналасқан виртуалды жады облысы
2. Берілген процесте орындалатын программаның инициализацияланған және инициализацияланбаған статикалық айнымалылардан тұрады
3. Тек қана командалардан тұрады және процестің орындалу барысында жаңара алмайды
4. Бөлінетін жадының сегментінің виртуалды жадыға қосылуы кезінде пайда болады
5. Бөлінетін сегменттердің түрлерінен құралады және оның беттері қажеттілік кезінде файл орналасқан ішкі жады облысында өз орнына жылжиды

67. UNIX ОЖ –нің виртуалды жадысының стек сегменті

1. Бөлінетін жадының сегментінің виртуалды жадыға қосылуы кезінде пайда болады
2. Бөлінетін сегменттердің түрлерінен құралады және оның беттері қажеттілік кезінде файл орналасқан ішкі жады облысында өз орнына жылжиды
3. Программаның автоматты айнымалылары анық және анық емес орналасқан виртуалды жады облысы
4. Берілген процесте орындалатын программаның инициализацияланған және инициализацияланбаған статикалық айнымалылардан тұрады

5. Тек қана командалардан тұрады және процестің орындалу барысында жаңара алмайды

68. UNIX ОЖ –нің виртуалды жадысының файлдар сегменті

1. Бөлінетін сегменттердің түрлерінен құралады және оның беттері қажеттілік кезінде файл орналасқан ішкі жады облысында өз орнына жылжиды
2. Бөлінетін жадының сегментінің виртуалды жадыға қосылуы кезінде пайда болады
3. Программаның автоматты айнымалылары анық және анық емес орналасқан виртуалды жады облысы
4. Берілген процесте орындалатын программаның инициализацияланған және инициализацияланбаған статикалық айнымалылардан тұрады
5. Тек қана командалардан тұрады және процестің орындалу барысында жаңара алмайды

69. UNIX ОЖ –нің физикалық оперативті жадысының қанша беттер тізімі бар?

1. 3
2. 2
3. 4
4. 5
5. 6

70. UNIX ОЖ –нің физикалық оперативті жадысының бірінші беттер тізімі қандай ақпараттардан құралады?

1. Ол өзіне беттерді сипаттаушыны қосады, ол модификацияны орындамайды немесе қандай да бір файлдың ішкі жады облысына сипаттайды. Бұндай беттер үшін толтырылған жүйе облысында кеңістік қажет етілмейді
2. Бұл бос беттер сипаттаушыларының тізімі
3. Ол өзіне жасырын деп аталатын беттерді сипаттаушыны қосады, яғни бұл өзгере алатын бірақ ішкі жадыда «туған» орны болмайтын беттер
4. Ол өзіне беттерді сипаттаушыны қосады, ол модификацияны орындайды немесе қандай да бір файлдың ішкі жады облысына сипаттайды
5. Бұл толтырылған беттер сипаттаушыларының тізімі

71. UNIX ОЖ –нің физикалық оперативті жадысының екінші беттер тізімі қандай ақпараттардан құралады?

1. Бұл бос беттер сипаттаушыларының тізімі
2. Ол өзіне беттерді сипаттаушыны қосады, ол модификацияны орындамайды немесе қандай да бір файлдың ішкі жады облысына сипаттайды. Бұндай беттер үшін толтырылған жүйе облысында кеңістік қажет етілмейді
3. Ол өзіне жасырын деп аталатын беттерді сипаттаушыны қосады, яғни бұл өзгере алатын бірақ ішкі жадыда «туған» орны болмайтын беттер
4. Ол өзіне беттерді сипаттаушыны қосады, ол модификацияны орындайды немесе қандай да бір файлдың ішкі жады облысына сипаттайды
5. Бұл толтырылған беттер сипаттаушыларының тізімі

72. UNIX ОЖ –нің физикалық оперативті жадысының үшінші беттер тізімі қандай ақпараттардан құралады?

1. Ол өзіне жасырын деп аталатын беттерді сипаттаушыны қосады, яғни бұл өзгере алатын бірақ ішкі жадыда «туған» орны болмайтын беттер

2. Ол өзіне беттерді сипаттаушыны қосады, ол модификацияны орындамайды немесе қандай да бір файлдың ішкі жады облысына сипаттайды. Бұндай беттер үшін толтырылған жүйе облысында кеңістік қажет етілмейді
3. Бұл бос беттер сипаттаушыларының тізімі
4. Ол өзіне беттерді сипаттаушыны қосады, ол модификацияны орындайды немесе қандай да бір файлдың ішкі жады облысына сипаттайды
5. Бұл толтырылған беттер сипаттаушыларының тізімі

73. UNIX ОЖ – де диспетчерлеудің қандай түрі қолданылады?

1. Приоритет пен кванттауды қолдануға негізделген ығыстырушы көпміндеттілік
2. Приоритет пен кванттауды қолдануға негізделген ығыстырмайтын көпміндеттілік
3. Приоритетті қолдануға негізделген ығыстырушы көпміндеттілік
4. Кванттауды қолдануға негізделген ығыстырушы көпміндеттілік
5. Приоритетті көпміндеттілік

74. UNIX ОЖ – де қандай приоритеттер класы қолданылады?

1. Нақты уақыттағы, жүйелік процестер, жіктелген уақыт процестері
2. Жүйелік процестер, жіктелген уақыт процестері
3. Нақты уақыттағы, жіктелген уақыт процестері
4. Нақты уақыттағы, жүйелік процестер,
5. Жіктелген уақыт процестері

75. UNIX ОЖ – де қандай процестер тек терілген приоритеттер стратегиясын қолданады?

1. Жүйелік класс процестері
2. Жүйелік класс процестері және уақыттың бөлінуі
3. Нақты уақыт процестері
4. Нақты уақыт процестері және жүйелік процестер
5. Уақыттың бөліну процестері

76. UNIX ОЖ – де қандай процестер тек динамикалық приоритеттер стратегиясын қолданады?

1. Уақыттың бөліну процестері
2. Жүйелік класс процестері және уақыттың бөлінуі
3. Нақты уақыт процестері
4. Нақты уақыт процестері және жүйелік процестер
5. Жүйелік класс процестері

77. UNIX ОЖ – ң барлық процестері үшін квант уақытының бірдей өлшемі сәйкес келеді деген ұғым дұрыс па?

1. Жоқ
2. Иә

78. UNIX ОЖ – ң процесс контексті қандай құраушылардан тұрады?

1. Тұтынушылық, регистрлі, жүйелілік
2. Тұтынушылық, жүйелілік
3. Тұтынушылық, регистрлі,
4. Регистрлі, жүйелілік
5. Тұтынушылық

79. Windows NT үшін шындық деген не?

1. Ол қазіргі заманғы ОЖ – ға қойылатын барлық талаптарды қанағаттандырады
2. Preemption міндеттері

3. Ядро артықшыланған режимде жұмыс істейді және ешқашан жадыдан жойылмайды
 4. Windows NT Preemption ядросы
 5. Жүйені дисплей мен перне тақтасыз қолдануға болады
80. Windows NT ОЖ – де диспетчерлеудің қандай түрі қоланылады?
1. Приоритет пен кванттауды қолдануға негізделген ығыстырушы көпміндеттілік
 2. Приоритетті қолдануға негізделген ығыстырушы көпміндеттілік
 3. Приоритет пен кванттауды қолдануға негізделген ығыстырмайтын көпміндеттілік
 4. Кванттауды қолдануға негізделген ығыстырушы көпміндеттілік
 5. Приоритетті көпміндеттілік
81. Windows NT ОЖ – де құрылымдық жобалаудың қандай технологиялары қолданылады?
1. клиент-сервер, микроядро, объектілер
 2. клиент-сервер, микроядро
 3. клиент-сервер, микроядро, модульді құрылым
 4. жойылған процедуралар шақыру құрылғылары, микроядро, объектілер
 5. клиент-сервер, объектілер, жойылған процедуралар шақыру құрылғылары, модульді құрылым
82. Windows NT есептеуіш процестің асиметриялы көпроцессорлы ұйымын қабылдай ма?
1. жоқ
 2. иә
83. Windows NT-да көп процессорлы өңдеу қандай мақсаттар үшін қажет?
1. өндіріс оптимизациясы үшін
 2. тұрақтылықтан бас тартудың жоғарылауы үшін
 3. өндіріс оптимизациясы мен тұрақтылықтан бас тартудың жоғарылауы үшін
 4. ауыстырымдылық жоғарылауы үшін
 5. үйлесімділіктің жоғарылауы үшін
84. Windows NT-да қандай басты режимдер бар?
1. тұтынушы мен ядро режимі
 2. ядро режимі
 3. тұтынушы режимі де ядро режимі де
 4. HAL режимі
 5. HAL режимі, тұтынушы мен ядро режимі
85. Windows NT-да программалар сервистері қандай режимде орындалады?
1. тұтынушы режимі
 2. ядро режимі
 3. тұтынушы режимі де ядро режимі де
 4. HAL режимі
 5. HAL режимі, тұтынушы мен ядро режимі
86. Windows NT-да программалар сервистері қандай режимде орындалады?
1. ядро режимі

2. тұтынушы режимі
3. тұтынушы режимі де ядро режимі де
4. HAL режимі
5. HAL режимі, тұтынушы мен ядро режимі

87. Windows NT-ның серверлері мен сервистері арасындағы өзара әсерлесу процесі қалай орындалады?

1. хаттар көмегімен
2. семафорлар көмегімен
3. монитор көмегімен
4. семафорлар мен монитор көмегімен
5. семафорлар, монитор, хаттар көмегімен

88. Windows NT-ның енгізу/шығару шағын жүйесінің басты қызметі-

1. жойылған сұрауларды енгізу/шығаруға, машиналық желіге жіберетін, олардан сұраулар алатын файлдық жүйелер драйверлері, енгізу/шығару құрылғыларын ұсыну
2. Windows NT-ны бір платформадан екіншісіне ауыстыруды жеңілдету үшін машиналы-тәуелді процедуралардың үлкен бөлігін бір жерде концентрациялау
3. басқа операциялық жүйелердің қосымша интерфейстерін іске асыру
4. ОЖ-ның тасымалды басты бөлігі- тұтынушының орта жиынтығы қолдау құрылғылары ретінде қызмет ету, бірақ оны жеке қолданыс үшін бөліп алуға болады
5. виртуальды жадыны, объектілерді басқару, процестердің өзара әсерлерін қамтамасыз ету, жүйе қауіпсіздігін қамтамасыз ету.

89. Windows NT-ның аппаратты абстракция деңгейінің басты қызметі

1. Windows NT-ны бір платформадан екіншісіне ауыстыруды жеңілдету үшін машиналы-тәуелді процедуралардың үлкен бөлігін бір жерде концентрациялау
2. басқа операциялық жүйелердің қосымша интерфейстерін іске асыру
3. процестерді жоспарлау, үзілулер мен ерекше жағдайларды өңдеу, көп процессорлы жүйелерге арналған процессорларды синхрондау, ақаулардан кейін жүйені қалпына келтіру
4. виртуальды жадыны, объектілерді басқару, процестердің өзара әсерлерін қамтамасыз ету, жүйе қауіпсіздігін қамтамасыз ету.
5. жойылған сұрауларды енгізу/шығаруға, машиналық желіге жіберетін, олардан сұраулар алатын файлдық жүйелер драйверлері, енгізу/шығару құрылғыларын ұсыну

90. Windows NT-ның жүйелік сервистерінің басты қызметі

1. виртуальды жадыны, объектілерді басқару, процестердің өзара әсерлерін қамтамасыз ету, жүйе қауіпсіздігін қамтамасыз ету.
2. Windows NT-ны бір платформадан екіншісіне ауыстыруды жеңілдету үшін машиналы-тәуелді процедуралардың үлкен бөлігін бір жерде концентрациялау
3. басқа операциялық жүйелердің қосымша интерфейстерін іске асыру
4. ОЖ-ның тасымалды басты бөлігі- тұтынушының орта жиынтығы қолдау құрылғылары ретінде қызмет ету, бірақ оны жеке қолданыс үшін бөліп алуға болады
5. жойылған сұрауларды енгізу/шығаруға, машиналық желіге жіберетін, олардан сұраулар алатын файлдық жүйелер драйверлері, енгізу/шығару құрылғыларын ұсыну

91. Windows NT-ның серверлерінің (қорғалған шағын жүйелердің) басты қызметі
1. басқа операциялық жүйелердің қосымша интерфейстерін іске асыру
 2. Windows NT-ны бір платформадан екіншісіне ауыстыруды жеңілдету үшін машиналы-тәуелді процедуралардың үлкен бөлігін бір жерде концентрациялау
 3. процестерді жоспарлау, үзілулер мен ерекше жағдайларды өңдеу, көп процессорлы жүйелерге арналған процессорларды синхрондау, ақаулардан кейін жүйені қалпына келтіру
 4. виртуальды жадыны, объектілерді басқару, процестердің өзара әсерлерін қамтамасыз ету, жүйе қауіпсіздігін қамтамасыз ету.
 5. жойылған сұрауларды енгізу/шығаруға, машиналық желіге жіберетін, олардан сұраулар алатын файлдық жүйелер драйверлері, енгізу/шығару құрылғыларын ұсыну
92. Windows NT-ның микроядросының басты қызметі
1. процестерді жоспарлау, үзілулер мен ерекше жағдайларды өңдеу, көп процессорлы жүйелерге арналған процессорларды синхрондау, ақаулардан кейін жүйені қалпына келтіру
 2. Windows NT-ны бір платформадан екіншісіне ауыстыруды жеңілдету үшін машиналы-тәуелді процедуралардың үлкен бөлігін бір жерде концентрациялау
 3. басқа операциялық жүйелердің қосымша интерфейстерін іске асыру
 4. виртуальды жадыны, объектілерді басқару, процестердің өзара әсерлерін қамтамасыз ету, жүйе қауіпсіздігін қамтамасыз ету.
 5. жойылған сұрауларды енгізу/шығаруға, машиналық желіге жіберетін, олардан сұраулар алатын файлдық жүйелер драйверлері, енгізу/шығару құрылғыларын ұсыну
93. Windows NT-ның барлық жүйелерін біртұтас қылып біріктіретін басты құрылғы не?
1. локальді процедуралардың шақыру механизмі
 2. жойылған шақыру процедураларының механизмі
 3. аппаратты абстракция деңгейіндегі механизм
 4. желілік сервер
 5. объектілер менеджері
94. Windows NT операциялық жүйенің қанша қолданбалы ортасын қолданады?
- 1.5
 - 2.4
 - 3.3
 - 4.6
 - 5.2
95. Windows NT объектісі қандай басты бөліктерден тұрады?
1. тақырыпша
 2. дене
 3. қауіпсіздік дескрипторы
 4. аты
 5. сілтеуші
96. Windows NT-да объектінің тақырып форматы уникальды болып келе ма?
1. жоқ
 2. иә

97. Windows NT-да процестер қалай іске асырылған?
- 1.объектілер формасында және оларға рұқсат объектінің қызметі арқылы іске асырылады
 - 2.арнай файлдар арқылы, оларға рұқсат енгізу/шығару менеджері арқылы іске асырылады
 - 3.процедуралар формасында, оларға рұқсат локальді процедуралар құрылғылары арқылы іске асырылады
 - 4.қорғалған жүйелер формасында , оларға рұқсат жүйелік сервистер арқылы іске асырылады
 - 5.жүйелік сервистер формасында ,оларға аппаратты абстракция деңгейі арқылы іске асырылады
98. Windows NT-да процестер мен тізбектер үшін шындық деген не?
- 1.процесс көп тізбекті ұйымнан тұрады
 - 2.процестер де, тізбектер секілді ендірілген синхрондау құралдарынан тұрады
 - 3.түтынушы процестері ядро функциясына жүйелік шақырулар арқылы қол жеткізеді
 - 4.процестер менеджері процестер арасында «ата-ұрпақ» қатынасын сақтайды
 - 5.әрбір процестің адресі кеңістігі басқа процестің кедергісінен сақталмаған
99. Windows NT-ның процестері кез келген процессорда орындала ала ма?
- 1.жоқ
 - 2.иә
100. Windows NT-ның процесінің базалық приоритеті деген не?
- 1.процесс тізбегінің орындалу приоритеті үшін негіз
 - 2.процестің орындалушы приоритеті үшін негіз
 - 3.процестің динамикалық приоритетінің төменгі шекарасы
 - 4.процестің динамикалық приоритетінің жоғарғы шекарасы
 - 5.процес тізбегінің динамикалық приоритетінің жоғарғы шекарасы
101. Windows NT-ның тізбегінің базалық приоритеті деген не?
- 1.тізбектің динамикалық приоритетінің төменгі шекарасы
 - 2.процесс тізбегінің орындалу приоритеті үшін негіз
 - 3.процестің орындалушы приоритеті үшін негіз
 - 4.процестің динамикалық приоритетінің жоғарғы шекарасы
 - 5.процес тізбегінің динамикалық приоритетінің жоғарғы шекарасы
102. Windows NT-да қанша приоритет класы бар?
- 1.32
 - 2.127
 - 3.16
 - 4.255
 - 5.64
103. Windows NT-да қолданылатын приоритет деңгейлері қандай басты кластарға бөлінеді?
- 1.нақты уақыттағы және айнымалы приоритет
 - 2.нақты уақыттағы және жүйелік
 - 3.жүйелік және айнымалы приоритет
 - 4.жүйелік және уақыттың бөлінуі
 - 5.жүйелік, уақыттың бөлінуі және айнымалы приоритет

104. Windows NT-да интерактивті есептер үшін қандай жоспарлау жүргізілген?
- 1.адаптивті жоспарлау
 - 2.статикалық приоритетті қолдану арқылы жоспарлау
 - 3.приоритет күту уақытына тәуелді
 - 4.кооперативті кқпміндеттілік
 - 5.жоспарлаудың сызықты алгоритмі
105. Windows NT-дағы кейінгі шақыру процедуралары бірдей приоритетті деңгейден тұра ма?
- 1.иә
 - 2.жоқ
106. Windows NT-дағы үзілулерді өңдеу программасы қанша кезеңнен тұрады?
- 1.2
 - 2.3
 - 3.4
 - 4.1
 - 5.5
107. Windows NT-ның қандай қасиеттері оны НУОЖ ретінде қолдануға мүмкіндік береді?
- 1.preemption міндеттері
 - 2.preemption үзілулері
 - 3.preemptive жүйе ядросы
 - 4.нақты уақыт класындағы приоритеттің үлкен көлемі
 - 5.кейінге қалдырылған шақыру процедураларының механизмінің артықшылықтары
108. Windows NT-ның қандай қасиеттері оны НУОЖ ретінде қолдануға мүмкіндік бермейді?
- 1.preemptive емес жүйе ядросы
 - 2.нақты уақыт класындағы приоритеттің аз көлемі
 - 3.кейінге қалдырылған шақыру процедураларының механизмінің кемшіліктері
 - 4.preemption емес міндеттері
 - 5.preemption емес үзілулері
109. Нақты уақыттағы операциялық жүйелердің ядролары Windows NT ядроларын қосымша ретінде қолдану кезінде Windows NT-ның құралуына мүмкіндік береді және ендірмелі құрам құрады деген тұжырым дұрыс па?
- 1.жоқ
 - 2.иә

СОДЖ бойынша тесттер

1. ОЖ-ның даму кезеңі неше кезеңнен тұрады?
 1. 7
 2. 6
 3. 5
 4. 4
 5. 3
2. ОЖ-ның қандай даму периодында персоналдың жіктелінуі жүргізіледі?
 1. 2
 2. 3
 3. 4

4. 5
5. 1
3. ОЖ-ның қандай даму периодында бірінші алгоритм тілдері мен программалық компиляторлар пайда болды?
 1. 4
 2. 3
 3. 2
 4. 1
 5. 5
4. ОЖ-ның қандай даму периодында мультипрограммалау іске асырылды?
 1. 2
 2. 4
 3. 5
 4. 1
 5. 3
5. Қандай аппаратты платформада ОЖ-нің 4 кезеңі пайда болды?
 1. интегралды микросхемалар
 2. жартылай өткізгішті элементтер
 3. транзисторлар
 4. үлкен интегралды сызбалар
 5. жартылай өткізгішті элементтер мен транзисторлар
6. Жүйелік программалық қамсыздандыру қандай басты группаларға бөлінеді?
 1. операциялық жүйелер, файлдарды басқару жүйесі, интерфейсті қабықшалар, программалау жүйесі, утилиттер, трансляторлар, шағын программалар кітапханасы
 2. операциялық жүйелер, файлдарды басқару жүйесі, программалау жүйесі, шағын программалар кітапханасы, трансляторлар, қолданбалы программалар пакеті
 3. операциялық жүйелер, файлдарды басқару жүйесі, МБ-н басқару жүйесі, интерфейсті қабықшалар, программалау жүйесі, редакторлар
 4. операциялық жүйелер, файлдарды басқару жүйесі, программалау жүйесі
 5. операциялық жүйелер, файлдарды басқару жүйесі, интерфейсті қабықшалар, программалау жүйесі, утилиттер
7. Әрбір операциялық жүйе бір ғана операциялық ортадан тұрады деген ұғым дұрыс па?
 1. жоқ
 2. иә
8. Кез келген жүйе өзіне сай құрылған ОЖ-де ғана жұмыс істей алады деген тұжырым дұрыс па?
 1. жоқ
 2. иә
9. Кейбір ОЖ бірнеше файлдық жүйелермен жұмыс істей алады деген ұғым дұрыс па?
 1. иә
 2. жоқ
10. Кейбір қарапайым ОЖ файлдық жүйелерсіз жұмыс істейді деген ұғым дұрыс па?
 1. иә
 2. жоқ