



ऑपरेटिंग सिस्टम — प्रश्न बैंक

BTEUP CSE, तीसरा सेमेस्टर (विषय 3.1) • उत्तर सहित प्रश्न

KAMALDNP kamaldnp.com

इस प्रश्न बैंक के बारे में — पहले यह पढ़िए

किसने बनाया. यह प्रश्न बैंक **KAMALDNP**, वाराणसी द्वारा बनाया गया है — एक IT प्रशिक्षण संस्थान जो कंप्यूटर पाठ्यक्रम और ग्रीष्मकालीन प्रशिक्षण देता है और डिप्लोमा छात्रों के लिए अध्ययन सामग्री प्रकाशित करता है। इसे संस्थान के शिक्षकों ने आधिकारिक BTEUP पाठ्यक्रम, साथी पाठ्यपुस्तक, और बोर्ड के अपने पिछले परीक्षा पत्रों से तैयार किया है।

क्यों बनाया. छात्र आमतौर पर बोर्ड के प्रश्नों से पहली बार परीक्षा हॉल के अंदर ही मिलते हैं। पाठ्यक्रम बताता है कि क्या पढ़ना है, लेकिन केवल बोर्ड के असली पत्र बताते हैं कि कैसे पूछा जाता है — कौन से विषय साल दर साल लौटते हैं, कौन से आंकिक प्रश्न पसंदीदा हैं, और 6 अंक के उत्तर में वास्तव में कितना लिखना पड़ता है। यह बैंक उसी आश्चर्य को हटाने के लिए है: यहाँ का अभ्यास असली पत्र का पूर्वाभ्यास लगे।

अंदर क्या है।

- **इकाई अनुसार प्रश्न उत्तर सहित** — पाठ्यक्रम की सभी पाँच इकाइयों के लिए उत्तर कुंजी सहित वस्तुनिष्ठ (MCQ) प्रश्न, मॉडल उत्तरों सहित लघु उत्तर प्रश्न, और वे बिंदु सहित दीर्घ उत्तर प्रश्न जो पूर्ण उत्तर में होने चाहिए।
- **पूरी तरह हल किए गए आंकिक प्रश्न** — CPU शेड्यूलिंग, बैंकर एल्गोरिदम, मेमोरी fit, पेजिंग, पेज प्रतिस्थापन और डिस्क शेड्यूलिंग, चरण दर चरण हल किए गए, पाठ्यपुस्तक के अध्यायों के अंत में छोटे अभ्यास सेटों के हल सहित।
- **मॉडल पेपर (पेपर सेट)** — बोर्ड के पैटर्न (60 अंक, 3 घंटे) पर संपूर्ण अभ्यास पत्र, उत्तर संकेतों सहित, ताकि आप परीक्षा जैसी स्थिति में पूर्वाभ्यास कर सकें।

- **पिछले वर्षों के प्रश्न, उनके साल के साथ** — बोर्ड के पिछले पत्रों के असली प्रश्न उस साल के साथ शामिल हैं जिसमें वे पूछे गए थे (उदाहरण के लिए, "2024 में पूछा गया"), 2020 से आगे के पत्रों से एकत्र किए गए। जहाँ विषय किसी पिछले साल किसी और नाम से या किसी और सेमेस्टर में आया था, वह नोट किया गया है — प्रश्न बोर्ड के अपने हैं, इसलिए दोहराए जाने वाले विषय ईमानदारी से खुद दिखते हैं।

कहाँ से और कैसे डाउनलोड करें। यह प्रश्न बैंक और इसकी साथी पाठ्यपुस्तक, ऑपरेटिंग सिस्टम, **kamaldnp.com** पर उपलब्ध हैं। उन्हें पाने के लिए: अपने browser में kamaldnp.com खोलिए, अध्ययन सामग्री / किताबों के खंड में जाइए, Operating System (BTEUP CSE 3rd Semester) खोजिए, और PDF डाउनलोड कीजिए — पाठ्यपुस्तक और प्रश्न बैंक अलग-अलग PDF फाइलों के रूप में दिए गए हैं, अपने फोन या कंप्यूटर पर सहेजने और छापने के लिए निःशुल्क। यदि कोई link कभी काम न करे, तो **support@kamaldnp.com** पर लिखिए।

इसे क्यों पढ़ना चाहिए। क्योंकि यह बोर्ड के अपने व्यवहार से बना है: पाठ्यक्रम क्षेत्र तय करता है, पिछले पत्र आदतें उजागर करते हैं, और मॉडल उत्तर अपेक्षित गहराई दिखाते हैं। जिस छात्र ने ईमानदारी से इस बैंक को पूरा किया है, वह असली पत्र के अधिकांश प्रश्नों — या उनके करीबी रिश्तेदारों — से कम से कम एक बार पहले ही मिल चुका है।

यह क्या नहीं है। यह बैंक साथी है, विकल्प नहीं। यह पाठ्यपुस्तक (जहाँ हर विषय शुरुआत से समझाया गया है), आपकी कक्षा के नोट, या कागज़ पर आपके अपने अभ्यास की जगह नहीं लेता। यहाँ छपे उत्तर मॉडल उत्तर हैं — परीक्षा में उन्हें अपने शब्दों और चित्रों में लिखिए। और यदि आपका पाठ्यक्रम या योजना विषय 3.1 (CSE, तीसरा सेमेस्टर) से अलग है, तो पहले अपना विषय कोड सुनिश्चित कीजिए; अन्य विषयों के प्रश्न बैंक अलग से प्रकाशित होते हैं।

इसका उपयोग कैसे करें। पाठ्यपुस्तक में एक इकाई समाप्त कीजिए, फिर उत्तर ढककर यहाँ उस इकाई को कीजिए। जाँचिए, सुधारिए, और आंकिक प्रश्नों को कागज़ पर तब तक दोहराइए जब तक विधि स्वतः न हो जाए। परीक्षा से पहले के अंतिम सप्ताहों में, पिछले वर्षों के प्रश्नों और मॉडल पेपर पर आ जाइए, उन्हें समय बाँधकर हल करते हुए, ठीक हॉल की तरह।

KAMALDNP द्वारा प्रकाशित · kamaldnp.com · support@kamaldnp.com



प्रश्न बैंक — इकाई 1: ऑपरेटिंग सिस्टम का परिचय

(BTEUP CSE तीसरा सेमेस्टर · विषय 3.1 · पेपर में इकाई का भार: लगभग 14%)

खंड A — वस्तुनिष्ठ प्रश्न (MCQ)

प्र1. ऑपरेटिंग सिस्टम मुख्य रूप से है:

- (a) एक एप्लिकेशन प्रोग्राम
- (b) उपयोगकर्ता और हार्डवेयर के बीच का सिस्टम सॉफ्टवेयर
- (c) CPU का एक भाग
- (d) एक प्रोग्रामिंग भाषा

उत्तर: (b)

प्र2. इनमें से कौन सा संसाधन OS द्वारा प्रबंधित नहीं होता?

- (a) CPU
- (b) मेमोरी
- (c) डिस्क
- (d) दस्तावेज़ में उपयोगकर्ता द्वारा टाइप किया गया डेटा

उत्तर: (d)

प्र3. बैच सिस्टम में काम:

- (a) उपयोगकर्ता की लगातार बातचीत के साथ चलते हैं
- (b) समूह बनाकर बिना बातचीत के एक के बाद एक चलते हैं
- (c) कई कंप्यूटरों पर चलते हैं
- (d) गारंटीशुदा समय-सीमा के साथ चलते हैं

उत्तर: (b)

प्र4. मल्टीप्रोग्रामिंग का मुख्य उद्देश्य है:

- (a) बेहतर ग्राफ़िक्स
- (b) I/O इंटज़ार के दौरान CPU को दूसरे काम पर लगाकर उसे व्यस्त रखना
- (c) एक समय में एक ही काम चलाना
- (d) कंप्यूटरों को नेटवर्क में जोड़ना

उत्तर: (b)

प्र5. टाइम-शेयरिंग का सबसे अच्छा वर्णन है:

- (a) एक मशीन पर एक समय में एक उपयोगकर्ता
- (b) तेज़ CPU बदलाव द्वारा कई उपयोगकर्ता एक मशीन साझा करते हैं
- (c) इंटरनेट पर फाइलें साझा करना
- (d) कई CPU वाला सिस्टम

उत्तर: (b)

प्र6. कई CPU के सच में समानांतर काम करने वाले सिस्टम को कहते हैं:

- (a) बैच
- (b) टाइम-शेयरिंग
- (c) मल्टीप्रोसेसिंग
- (d) एम्बेडेड

उत्तर: (c)

प्र7. हार्ड रियल-टाइम सिस्टम वह है जहाँ:

- (a) जवाब आमतौर पर तेज़ होते हैं
- (b) समय-सीमा चूकना असफलता है
- (c) ग्राफ़िक्स उच्च गुणवत्ता के होते हैं
- (d) कई उपयोगकर्ता साथ काम करते हैं

उत्तर: (b)

प्र8. वॉशिंग मशीन नियंत्रक के लिए कौन सा OS प्रकार उपयुक्त है?

- (a) डिस्ट्रिब्यूटेड
- (b) बैच
- (c) केवल रियल-टाइम
- (d) एम्बेडेड

उत्तर: (d)

प्र9. OS के दो मुख्य लक्ष्य हैं:

- (a) गति और रंग
- (b) सुविधा और कुशलता
- (c) सुरक्षा और गेमिंग
- (d) लागत और आकार

उत्तर: (b)

प्र10. किस पीढ़ी ने पहली बार कई कामों को मेमोरी में एक साथ रहने दिया?

- (a) कोई OS नहीं
- (b) बैच
- (c) मल्टीप्रोग्रामिंग
- (d) पर्सनल कंप्यूटर

उत्तर: (c)

प्र11. डिस्ट्रिब्यूटेड OS कई कंप्यूटरों को इस रूप में दिखाता है:

- (a) अलग-अलग मशीनें
- (b) एक ही सिस्टम
- (c) प्रिंटर नेटवर्क
- (d) बैच कतार

उत्तर: (b)

प्र12. इनमें से कौन एक ऑपरेटिंग सिस्टम है?

- (a) MS Word
- (b) Linux
- (c) Google Chrome
- (d) Tally

उत्तर: (b)

प्र13. टाइम-शेयरिंग सिस्टम में हर उपयोगकर्ता/प्रोग्राम को मिलता है:

- (a) घंटों के लिए पूरी मशीन
- (b) बारी-बारी से समय के छोटे टुकड़े
- (c) एक अलग CPU
- (d) केवल छपा हुआ आउटपुट

उत्तर: (b)

प्र14. सॉफ़्ट रियल-टाइम सिस्टम इस्तेमाल होते हैं:

- (a) पेसमेकर में
- (b) उड़ान नियंत्रण में
- (c) वीडियो स्ट्रीमिंग में
- (d) मिसाइल मार्गदर्शन में

उत्तर: (c)

प्र15. OS परत सीधे किसके साथ काम करती है:

- (a) केवल उपयोगकर्ता से
- (b) केवल एप्लिकेशन से
- (c) हार्डवेयर से, प्रोग्रामों और उपयोगकर्ताओं की ओर से
- (d) केवल इंटरनेट से

उत्तर: (c)

खंड B — लघु उत्तर प्रश्न (मॉडल उत्तर सहित)

प्र1. ऑपरेटिंग सिस्टम क्या है? ऑपरेटिंग सिस्टम वह सिस्टम सॉफ़्टवेयर है जो उपयोगकर्ता/एप्लिकेशन प्रोग्रामों और कंप्यूटर हार्डवेयर के बीच रहता है। यह पूरी मशीन को नियंत्रित करता है, उसके संसाधनों (CPU, मेमोरी, डिस्क, I/O उपकरण) का प्रबंधन करता है और प्रोग्रामों को चलने के लिए आवश्यक सेवाएँ और वातावरण प्रदान करता है। उदाहरण: Windows, Linux, Android।

प्र2. ऑपरेटिंग सिस्टम के दो लक्ष्य एक-एक उदाहरण सहित बताइए। सुविधा — उपयोगकर्ता हार्डवेयर के अंदर की बातें जाने बिना प्रोग्राम चला सके और फाइल सहेज सके (उदाहरण: फाइल पर डबल-क्लिक करने से वह खुल जाती है)। कुशलता — हार्डवेयर खाली न बैठे (उदाहरण: जब एक प्रोग्राम प्रिंटर का इंतज़ार करता है, CPU दूसरा प्रोग्राम चलाता है)।

प्र3. बैच ऑपरेटिंग सिस्टम क्या है? यह कहाँ उपयुक्त है? एक प्रकार के कामों को बैच में इकट्ठा करके, चलने के दौरान उपयोगकर्ता की बातचीत के बिना, एक के बाद एक चलाया जाता है। यह बड़े डेटा पर बार-बार दोहराई जाने वाली प्रक्रिया के लिए उपयुक्त है, जैसे पुराना वेतन और बिलिंग कार्य। इसकी सीमाएँ हैं: कोई बातचीत नहीं और धीमा प्रत्युत्तर समय।

प्र4. मल्टीप्रोग्रामिंग CPU का खाली समय कैसे घटाती है? कई काम मेमोरी में एक साथ रखे जाते हैं। जब चलता हुआ काम I/O का इंतज़ार करता है, तो OS CPU को खाली बैठने देने के बजाय दूसरे तैयार काम पर ले जाता है — इसलिए CPU लगभग हमेशा काम करता रहता है।

प्र5. मल्टीटास्किंग और टाइम-शेयरिंग में अंतर बताइए (दो बिंदु)। मल्टीटास्किंग: एक ही उपयोगकर्ता तेज़ CPU बदलाव द्वारा एक मशीन पर कई प्रोग्राम चलाता है। टाइम-शेयरिंग: कई उपयोगकर्ता टर्मिनलों द्वारा एक मशीन साझा करते हैं; यह मल्टीटास्किंग का बहु-उपयोगकर्ता विस्तार है, जो समय के टुकड़ों और हर उपयोगकर्ता के लिए त्वरित जवाब पर आधारित है।

प्र6. रियल-टाइम OS क्या है? इसके दो प्रकार बताइए। वह OS जो तय समय-सीमा के भीतर जवाब की गारंटी देता है; देर होना असफलता है। हार्ड रियल-टाइम कभी समय-सीमा नहीं चूकता (उड़ान नियंत्रण)। सॉफ़्ट रियल-टाइम कभी-कभार की देरी सह लेता है (वीडियो स्ट्रीमिंग)।

प्र7. एम्बेडेड OS क्या है? इसका उपयोग करने वाले दो उपकरणों के उदाहरण दीजिए। किसी उपकरण के अंदर बना छोटा, विशेष OS जो एक ही निश्चित काम के लिए होता है। उदाहरण: वॉशिंग मशीन नियंत्रक, ATM, स्मार्ट TV, माइक्रोवेव ओवन।

प्र8. ऑपरेटिंग सिस्टम के विकास को चार चरणों में लिखिए। (1) कोई OS नहीं — हाथ से संचालन, कामों के बीच मशीन खाली। (2) बैच — कामों का स्वतः क्रमिक चलना। (3) मल्टीप्रोग्रामिंग — I/O इंतज़ार के दौरान CPU का बदलाव। (4) टाइम-शेयरिंग — समय के टुकड़ों द्वारा कई उपयोगकर्ताओं का बातचीत युक्त उपयोग।

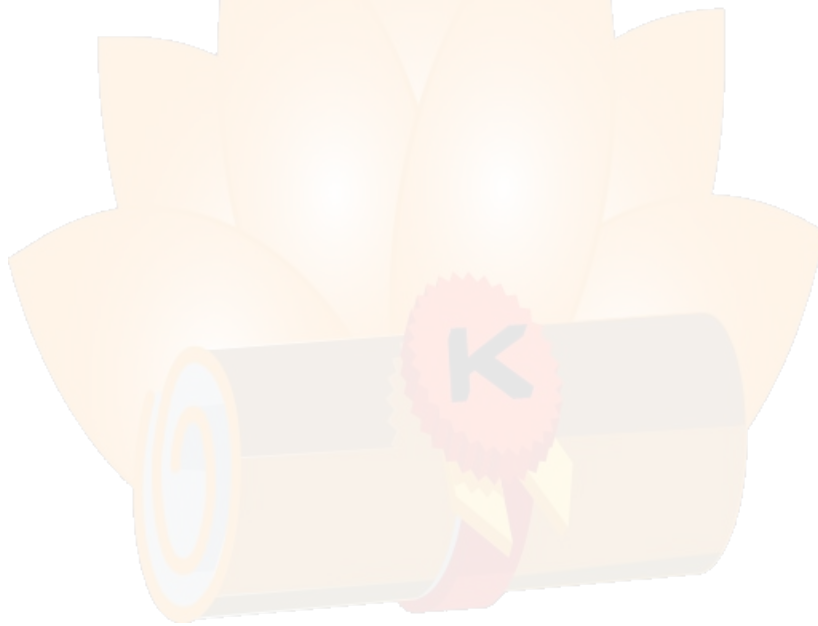
खंड C — दीर्घ उत्तर प्रश्न (मॉडल उत्तर सहित)

प्र1. ऑपरेटिंग सिस्टम को परिभाषित कीजिए। उपयोगकर्ता और हार्डवेयर के बीच इसकी स्थिति को चित्र वर्णन सहित समझाइए, और इसके दो लक्ष्यों पर विस्तार से चर्चा कीजिए। मॉडल उत्तर (विकसित करने के मुख्य बिंदु): परिभाषा (सिस्टम सॉफ़्टवेयर; इंटरफ़ेस/प्रबंधक परत) → परतदार चित्र: उपयोगकर्ता → एप्लिकेशन प्रोग्राम → OS → हार्डवेयर; अनुरोध नीचे जाते हैं, परिणाम ऊपर आते हैं → लक्ष्य 1, सुविधा: हार्डवेयर का ब्योरा छिपाता है, आसान सेवाएँ देता है (फाइल, प्रिंटिंग, प्रोग्राम चलाना) → लक्ष्य 2, कुशलता: संसाधन प्रबंधन ताकि CPU/मेमोरी/उपकरण शायद ही खाली रहें; मल्टीप्रोग्रामिंग इसका क्लासिक उदाहरण → निष्कर्ष: आगे की इकाइयों का हर OS कार्य इन्हीं दो लक्ष्यों में से एक की सेवा करता है।

प्र2. ऑपरेटिंग सिस्टम के मुख्य प्रकारों का हर एक के एक उदाहरण सहित वर्णन कीजिए। मॉडल उत्तर (मुख्य बिंदु): बैच (समूहित काम, कोई बातचीत नहीं; वेतन) → मल्टीप्रोग्राम्ड (काम मेमोरी में, I/O इंतज़ार पर CPU बदलाव) → मल्टीटास्किंग/टाइम-शेयरिंग (समय के टुकड़े, बातचीत, कई उपयोगकर्ता; कॉलेज सर्वर) → मल्टीप्रोसेसिंग (कई CPU समानांतर; सर्वर) → रियल-टाइम (समय-सीमा की गारंटी; हार्ड — उड़ान नियंत्रण, सॉफ़्ट — स्ट्रीमिंग) → डिस्ट्रिब्यूटेड (कई मशीनें एक जैसी; क्लस्टर) → एम्बेडेड (उपकरण के अंदर; ATM) → हर जोड़ी के लिए एक तुलना लाइन जो परीक्षक जोड़ सकता है (बातचीत है या नहीं; एक CPU या कई; समय-सीमा है या नहीं)।

प्र3. "टाइम-शेयरिंग सिस्टम मल्टीप्रोग्रामिंग का तार्किक विस्तार है।" इस कथन की पुष्टि कीजिए।
मॉडल उत्तर (मुख्य बिंदु): मल्टीप्रोग्रामिंग कई काम मेमोरी में रखती है और I/O इंतज़ार पर बदलती है — लेकिन बदलाव केवल तब होता है जब काम इंतज़ार करे, और उपयोगकर्ता बातचीत नहीं कर सकते → टाइम-शेयरिंग टाइमर (समय के टुकड़े) पर भी बदलती है, हर सेकंड कई बार, चाहे काम इंतज़ार करे या न करे → नतीजा: हर उपयोगकर्ता को बार-बार छोटे मोड़ मिलते हैं और लगातार ध्यान का एहसास होता है → वही मेमोरी-स्थित बहु-काम आधार, विस्तारित बदलाव नियम, नई विशेषता (बातचीत) — इसलिए यह तार्किक विस्तार है, कोई अलग आविष्कार नहीं।

प्र4. बैच, टाइम-शेयरिंग और रियल-टाइम सिस्टम की तुलना कीजिए: बातचीत, जवाब का बर्ताव, CPU उपयोग, और आम उपयोग के आधार पर। मॉडल उत्तर (तालिका रूप में): बातचीत — बैच: चलने के दौरान कोई नहीं; टाइम-शेयरिंग: पूरी; रियल-टाइम: डिज़ाइन के अनुसार, समय-सीमा से बँधा। जवाब — बैच: घंटों; टाइम-शेयरिंग: सेकंड या कम; रियल-टाइम: गारंटीशुदा तय सीमा। CPU उपयोग — बैच: अच्छा क्रम लेकिन लोड/अनलोड अंतराल में खाली; टाइम-शेयरिंग: बदलाव द्वारा उच्च; रियल-टाइम: समय-सीमाओं के लिए जितना ज़रूरी, अधिकतम प्रवाह के लिए नहीं। उपयोग — बैच: वेतन/बिलिंग; टाइम-शेयरिंग: कॉलेज सर्वर, डेस्कटॉप; रियल-टाइम: उड़ान नियंत्रण (हार्ड), स्ट्रीमिंग (सॉफ़्ट)।





प्रश्न बैंक — इकाई 2: प्रक्रिया प्रबंधन

(BTEUP CSE तीसरा सेमेस्टर · विषय 3.1 · पेपर में इकाई का भार: लगभग 25%)

खंड A — वस्तुनिष्ठ प्रश्न (MCQ)

प्र1. प्रक्रिया है:

- (a) डिस्क पर रखा प्रोग्राम
- (b) निष्पादन में चल रहा प्रोग्राम
- (c) कर्नेल का एक भाग
- (d) एक फाइल

उत्तर: (b)

प्र2. एक प्रोग्राम पाँच विंडो में खुलने से मिलता है:

- (a) 5 प्रोग्राम, 1 प्रक्रिया
- (b) 1 प्रोग्राम, 5 प्रक्रियाएँ
- (c) 5 प्रोग्राम, 5 प्रक्रियाएँ
- (d) 1 प्रोग्राम, 1 प्रक्रिया

उत्तर: (b)

प्र3. प्रतीक्षारत प्रक्रिया के लिए I/O पूरा होने पर, वह जाती है:

- (a) running में
- (b) terminated में
- (c) ready में
- (d) new में

उत्तर: (c)

प्र4. PCB क्या संग्रहीत नहीं करता:

- (a) प्रक्रिया अवस्था
- (b) प्रोग्राम काउंटर
- (c) प्रोग्राम का स्रोत कोड
- (d) CPU रजिस्टर

उत्तर: (c)

प्र5. कॉन्टेक्स्ट स्विच है:

- (a) लागत-मुक्त
- (b) शुद्ध overhead
- (c) प्रिंटिंग के लिए एक प्रकार की सिस्टम कॉल
- (d) उपयोगकर्ता द्वारा किया गया

उत्तर: (b)

प्र6. मल्टीप्रोग्रामिंग की मात्रा नियंत्रित होती है:

- (a) अल्पकालिक शेड्यूलर से
- (b) मध्यकालिक शेड्यूलर से
- (c) दीर्घकालिक शेड्यूलर से
- (d) केवल dispatcher से

उत्तर: (c)

प्र7. कौन सी शेड्यूलिंग डिज़ाइन से नॉन-प्रीएम्प्टिव है?

- (a) Round Robin
- (b) SRTF
- (c) FCFS
- (d) प्रीएम्प्टिव priority

उत्तर: (c)

प्र8. SJF सर्वोत्तम संभव देता है:

- (a) प्रत्युत्तर समय
- (b) औसत प्रतीक्षा समय
- (c) हमेशा श्रृंखला
- (d) निष्पक्षता

उत्तर: (b)

प्र9. काफ़िला प्रभाव (convoy effect) संबंधित है:

- (a) SJF से
- (b) RR से
- (c) FCFS से
- (d) SRTF से

उत्तर: (c)

प्र10. Round Robin में, यदि क्वांटम बहुत बड़ा हो जाए, तो वह इस प्रकार व्यवहार करता है:

- (a) SJF
- (b) FCFS
- (c) priority
- (d) LIFO

उत्तर: (b)

प्र11. Priority शेड्यूलिंग में भुखमरी का इलाज है:

- (a) बड़ा क्वांटम
- (b) एजिंग
- (c) स्पूलिंग
- (d) संकुचन

उत्तर: (b)

प्र12. रेस कंडीशन उत्पन्न होती है जब:

- (a) दो प्रक्रियाएँ अलग डेटा उपयोग करती हैं
- (b) परिणाम साझा डेटा तक पहुँच के क्रम पर निर्भर करता है
- (c) प्रक्रिया क्रैश होती है
- (d) CPU धीमा होता है

उत्तर: (b)

प्र13. कौन क्रिटिकल-सेक्शन हल की शर्त नहीं है?

- (a) परस्पर अपवर्जन
- (b) प्रगति
- (c) अधिकतम गति
- (d) सीमित प्रतीक्षा

उत्तर: (c)

प्र14. क्रिटिकल सेक्शन के चारों ओर उपयोग किया गया बाइनरी सेमाफोर यह भी कहलाता है:

- (a) मॉनिटर
- (b) mutex
- (c) pipe
- (d) inode

उत्तर: (b)

प्र15. कौन सी शर्त डेडलॉक के लिए आवश्यक नहीं है?

- (a) परस्पर अपवर्जन
- (b) धारण और प्रतीक्षा
- (c) प्रीएम्पशन अनुमत
- (d) वृत्तीय प्रतीक्षा

उत्तर: (c) — शर्त प्रीएम्पशन नहीं है।

खंड B — लघु उत्तर प्रश्न (मॉडल उत्तर सहित)

प्र1. प्रक्रिया बनाम प्रोग्राम — दो अंतर। प्रोग्राम डिस्क पर रखा निष्क्रिय कोड है; प्रक्रिया वह कोड है जो निष्पादन में है, अपनी मेमोरी, अवस्था और संसाधनों के साथ। एक प्रोग्राम एक साथ कई प्रक्रियाओं के रूप में चल सकता है; एक प्रक्रिया किसी प्रोग्राम के ठीक एक निष्पादन उदाहरण से संबंधित होती है।

प्र2. पाँच प्रक्रिया अवस्थाएँ गिनाइए और वह घटना बताइए जो running → ready ले जाती है। New, ready, running, waiting, terminated। Running → ready एक इंटरप्ट पर होता है — आमतौर पर समय टुकड़ा समाप्त होना — प्रक्रिया के अपने अनुरोध पर नहीं।

प्र3. मध्यकालिक शेड्यूलर क्या करता है? यह मेमोरी में भीड़ होने पर प्रक्रियाओं को मेमोरी से डिस्क पर swap out करता है (और बाद में वापस लाता है), दबाव कम करने के लिए मल्टीप्रोग्रामिंग की मात्रा अस्थायी रूप से घटाते हुए।

प्र4. टर्नअराउंड समय और प्रतीक्षा समय को सूत्रों सहित परिभाषित कीजिए। TAT = पूरा होने का समय – आगमन समय (सिस्टम में कुल जीवन)। WT = TAT – Burst समय (उस जीवन का वह भाग जो ready कतार में प्रतीक्षा में बीता)।

प्र5. पूर्ण SJF असली OS में असंभव क्यों है? OS किसी प्रक्रिया का भविष्य burst समय नहीं जान सकता; वह केवल पिछले burst के इतिहास से अगले burst का अनुमान लगा सकता है (चरघातांकी औसत)। इसलिए SJF लगभग लागू किया जाता है।

प्र6. सेमाफ़ोर पर wait और signal क्या हैं? Wait (P): सेमाफ़ोर को atomic रूप से घटाइए; परिणाम ऋणात्मक हो तो बुलाने वाले को रोकिए। Signal (V): atomic रूप से बढ़ाइए; प्रक्रियाएँ रुकी हों तो एक को जगाइए। उनकी atomicता ही उन्हें सुरक्षित रक्षक बनाती है।

प्र7. चार डेडलॉक शर्तों को चार छोटे वाक्यांशों में बताइए। परस्पर अपवर्जन (प्रति संसाधन एक धारक); धारण और प्रतीक्षा (कुछ धारण करो, और माँगो); कोई प्रीएम्पशन नहीं (संसाधन केवल स्वेच्छा से छूटते हैं); वृत्तीय प्रतीक्षा (प्रतीक्षारत प्रक्रियाओं का घेरा)।

प्र8. सुरक्षित, असुरक्षित, डेडलॉक — एक-एक लाइन। सुरक्षित: एक क्रम मौजूद है जो हर प्रक्रिया को उसके अधिकतम तक पहुँचकर समाप्त होने देता है। असुरक्षित: ऐसे क्रम की गारंटी नहीं है — डेडलॉक संभव है लेकिन हुआ नहीं है। डेडलॉक: प्रक्रियाओं का समूह पहले से ही प्रतीक्षा घेरे में स्थायी रूप से अवरुद्ध है।

खंड C — दीर्घ उत्तर प्रश्न (मॉडल उत्तर सहित)

प्र1. प्रक्रिया अवस्था चित्र बनाइए और हर बदलाव समझाइए। PCB की सामग्री का भी वर्णन कीजिए। मुख्य बिंदु: पाँच अवस्थाएँ; तीर — प्रवेश (new→ready), dispatch (ready→running), इंटरप्ट (running→ready), I/O प्रतीक्षा (running→waiting), I/O पूर्ति (waiting→ready, कभी सीधे running नहीं), exit (running→terminated)। PCB: PID, अवस्था, प्रोग्राम काउंटर, रजिस्टर, शेड्यूलिंग जानकारी, मेमोरी जानकारी, लेखा, I/O स्थिति — और कॉन्टेक्स्ट स्विच में उसकी भूमिका (पुराने PCB में सहेजना, नए PCB से लोड करना)।

प्र2. FCFS, SJF, Round Robin और Priority शेड्यूलिंग की नियम, प्रीएम्पशन, शक्ति और कमज़ोरी पर तुलना कीजिए। आप (i) बैच पेरोल सिस्टम (ii) बातचीत वाली लैब के लिए किसे चुनेंगे — और क्यों? मुख्य बिंदु: अध्याय 4 §4.6 की तुलना तालिका; चुनाव — पेरोल: SJF/FCFS स्वीकार्य (कोई बातचीत नहीं; औसत प्रतीक्षा के लिए SJF, सरलता/पूर्वानुमेयता के लिए FCFS); बातचीत लैब: Round Robin (प्रत्युत्तर समय और निष्पक्षता), संभवतः priority वर्गों के साथ; priority के लिए भुखमरी/एजिंग संबोधित; क्वांटम समझौता बताया गया।

प्र3. उत्पादक-उपभोक्ता समस्या और उसके तीन-सेमाफ़ोर हल की व्याख्या कीजिए; फिर दिखाइए कि डाइनिंग फ़िलॉसफ़र्स डेडलॉक कैसे उत्पन्न करते हैं और उसे कैसे रोका जाए। मुख्य बिंदु: सीमित बफ़र; नियम (भरने पर उत्पादक प्रतीक्षा करे, खाली होने पर उपभोक्ता, कभी दोनों अंदर नहीं); सेमाफ़ोर mutex=1, empty=N, full=0, wait क्रम सहित (mutex से पहले संसाधन सेमाफ़ोर);

फिलॉसफर्स: पाँच स्टिक, सब बाई स्टिक लेने पर वृत्तीय प्रतीक्षा बनती है; शर्त तोड़कर रोकथाम — दोनों स्टिक atomic रूप से लेना (धारण-और-प्रतीक्षा तोड़ना), क्रम नियम (वृत्तीय प्रतीक्षा तोड़ना), या अधिकतम चार बिठाना (निर्माण द्वारा वृत्तीय प्रतीक्षा तोड़ना)।

प्र4. चार डेडलॉक-सँभाल रणनीतियाँ बताइए। रोकथाम के लिए, प्रति शर्त एक विधि उसकी लागत सहित दीजिए। सामान्य-उद्देश्य OS अधिकतर डेडलॉक को क्यों अनदेखा करते हैं? मुख्य बिंदु: अनदेखा/ostrich (रोकथाम की लागत के मुकाबले दुर्लभता; पुनः प्रारंभ द्वारा पुनर्प्राप्ति); अध्याय 6 §6.3 की रोकथाम तालिका (परस्पर अपवर्जन के विरुद्ध स्पूलिंग; सब-कुछ-एक-साथ माँगना; CPU/मेमोरी-जैसे संसाधनों को प्रीएम्प्ट करना; संसाधन क्रम); परिहार (बैंकर को Max पहले से चाहिए — शायद ही उपलब्ध); पहचान और पुनर्प्राप्ति (पीड़ितों को समाप्त/प्रीएम्प्ट करना)। निष्कर्ष: सामान्य कार्यभार के लिए दुर्लभ डेडलॉक से अपेक्षित हानि रोकथाम की निश्चित लागत से छोटी है — एक इंजीनियरिंग समझौता, अनदेखी नहीं।

खंड D — हल किए गए आंकिक प्रश्न

N1. (अध्याय 4 अभ्यास सेट, हल किया गया।) प्रक्रियाएँ A (AT 0, BT 6), B (AT 1, BT 3), C (AT 2, BT 2), D (AT 4, BT 4)।

(a) FCFS: क्रम A, B, C, D। A 0–6; B 6–9; C 9–11; D 11–15। CT: 6, 9, 11, 15। TAT = CT – AT: 6, 8, 9, 11 → **औसत TAT 8.5**। WT = TAT – BT: 0, 5, 7, 7 → **औसत WT 4.75**।

(b) SJF (नॉन-प्रीएम्प्टिव): 0 पर A अकेली 6 तक चलती है। Ready: C (2), B (3), D (4) → C 6–8; B 8–11; D 11–15। CT: A 6, C 8, B 11, D 15। TAT: 6, 10, 6, 11 → **औसत 8.25**। WT: 0, 7, 4, 7 → **औसत 4.5**।

(c) SRTF: A शुरू होती है; 1 पर, B (3) < A का शेष 5 → B चलती है; 2 पर, C (2) B के शेष 2 से बराबर है (छोटा नहीं — कोई बदलाव नहीं); B 4 पर समाप्त। Ready: C (2), D (4), A (5) → C 4–6; D 6–10; A 10–15। CT: B 4, C 6, D 10, A 15। TAT: 15, 3, 4, 6 → **औसत 7**। WT: 9, 0, 2, 2 → **औसत 3.25** — तीनों में सर्वश्रेष्ठ, जैसा SRTF वादा करता है।

(d) Round Robin, क्वांटम 2: A 0–2 → कतार B, C, A → B 2–4 (D 4 पर आती है; कतार C, A, D, B) → C 4–6 समाप्त → A 6–8 → D 8–10 → B 10–11 समाप्त → A 11–13 समाप्त → D 13–15 समाप्त। CT: A 13, B 11, C 6, D 15। TAT: 13, 10, 4, 11 → **औसत 9.5**। WT: 7, 7, 2, 7 → **औसत 5.75**। ईमानदारी से नोट कीजिए: RR प्रत्युत्तर को अनुकूलित करता है (हर प्रक्रिया समय 4 तक चली), औसत प्रतीक्षा को नहीं — इस प्रतिरूप पर यह WT में चारों में सबसे खराब है, जो ठीक अध्याय 4 में वर्णित समझौता है।

N2. (अध्याय 4 अभ्यास सेट, हल किया गया।) अध्याय 4 का उदाहरण 2 समूह — P1 (AT 0, BT 8), P2 (AT 1, BT 4), P3 (AT 2, BT 9), P4 (AT 3, BT 5) — प्राथमिकताओं P1 = 2, P2 = 4, P3 = 1, P4 = 3 (छोटा = उच्च) के साथ **नॉन-प्रीएम्प्टिव priority** के अंतर्गत। P1 0 पर अकेली है और (नॉन-प्रीएम्प्टिव) 8 पर पूरी होती है। Ready प्राथमिकताएँ: P3 (1), P4 (3), P2 (4)। क्रम: P3 8-17; P4 17-22; P2 22-26। WT: P1 = 0; P3 = 17 - 2 - 9 = 6; P4 = 22 - 3 - 5 = 14; P2 = 26 - 1 - 4 = 21। **औसत WT = (0 + 6 + 14 + 21)/4 = 10.25।** अध्याय के उदाहरण 4 से तुलना कीजिए (प्राथमिकताएँ 3, 2, 4, 1 → औसत 8): एक समान काम, अलग प्राथमिकता नियतन, अलग पीड़ा — प्राथमिकता मान ही नीति हैं।

N3. (अध्याय 6 अभ्यास सेट, हल किया गया।) अध्याय 6 की हल अवस्था से (Available (3, 3, 2); P0 Need (7, 4, 3), Allocation (0, 1, 0)), **P0 (0, 2, 0) माँगती है।** जाँच 1: (0, 2, 0) ≤ Need (7, 4, 3) ✓। जाँच 2: ≤ Available (3, 3, 2) ✓। नाटक: Available → (3, 1, 2); P0 Allocation → (0, 3, 0); Need → (7, 2, 3)। Work (3, 1, 2) से सुरक्षा: P3 Need (0, 1, 1) बैठता है → Work (5, 2, 3); P1 (1, 2, 2) बैठता है → (7, 2, 3); P0 (7, 2, 3) बैठता है → (7, 5, 3); P4 (4, 3, 1) बैठता है → (7, 5, 5); P2 (6, 0, 0) बैठता है। सभी समाप्त होते हैं — **सुरक्षित, माँग मंजूर।** (और अध्याय का अन्य अभ्यास: P3, P1, P4, P2, P0 भी एक वैध सुरक्षित क्रम है — सुरक्षित अवस्था के आमतौर पर कई क्रम होते हैं।)





प्रश्न बैंक — इकाई 3: मेमोरी प्रबंधन

(BTEUP CSE तीसरा सेमेस्टर · विषय 3.1 · पेपर में इकाई का भार: लगभग 25%)

खंड A — वस्तुनिष्ठ प्रश्न (MCQ)

प्र1. तार्किक से भौतिक एड्रेस में अनुवाद किया जाता है:

- (a) कंपाइलर द्वारा
- (b) OS द्वारा रखे मानों का उपयोग करके MMU हार्डवेयर द्वारा
- (c) उपयोगकर्ता द्वारा
- (d) डिस्क कंट्रोलर द्वारा

उत्तर: (b)

प्र2. निश्चित विभाजन में, विभाजन के अंदर की बर्बादी है:

- (a) बाहरी विखंडन
- (b) आंतरिक विखंडन
- (c) संकुचन
- (d) थ्रैशिंग

उत्तर: (b)

प्र3. Best Fit चुनता है:

- (a) पहला छिद्र जो बैठता है
- (b) सबसे बड़ा छिद्र
- (c) सबसे छोटा छिद्र जो बैठता है
- (d) यादृच्छिक छिद्र

उत्तर: (c)

प्र4. स्वैपिंग किया जाता है:

- (a) दीर्घकालिक शेड्यूलर द्वारा
- (b) अल्पकालिक शेड्यूलर द्वारा
- (c) मध्यकालिक शेड्यूलर द्वारा
- (d) dispatcher द्वारा

उत्तर: (c)

प्र5. पेजिंग में, भौतिक मेमोरी ब्लॉक कहलाते हैं:

- (a) पेज
- (b) फ्रेम
- (c) सेगमेंट
- (d) क्लस्टर

उत्तर: (b)

प्र6. पेजिंग अनुवाद के दौरान, एड्रेस का कौन सा भाग कभी नहीं बदलता?

- (a) पेज संख्या
- (b) फ्रेम संख्या
- (c) ऑफ़सेट
- (d) बेस

उत्तर: (c)

प्र7. पेजिंग समाप्त करती है:

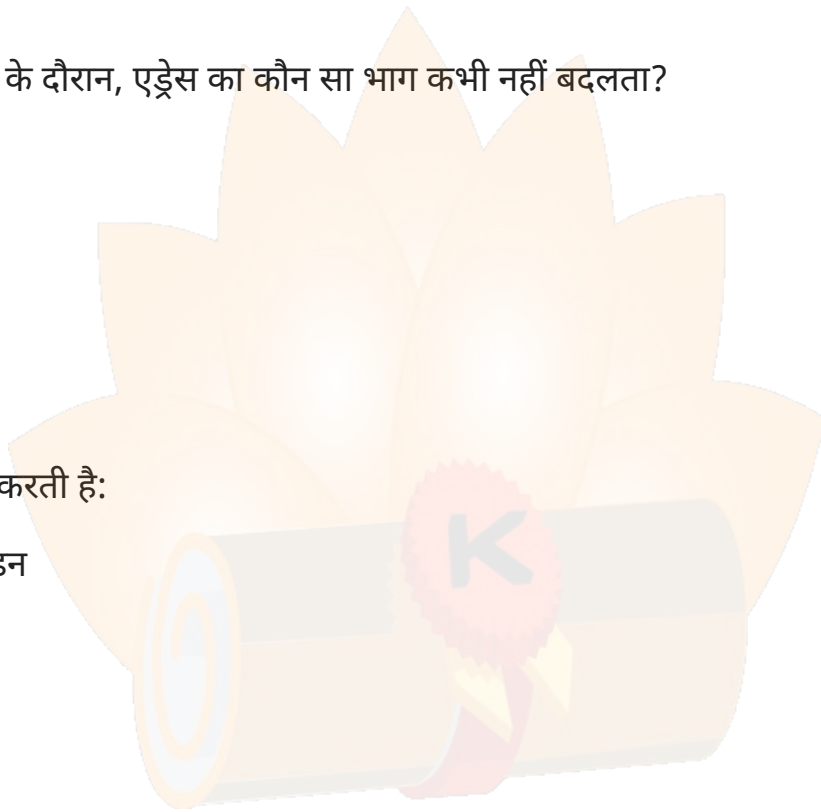
- (a) आंतरिक विखंडन
- (b) बाहरी विखंडन
- (c) पेज फ्रॉन्ट
- (d) पेज टेबल

उत्तर: (b)

प्र8. सेगमेंट टेबल प्रविष्टि रखती है:

- (a) केवल फ्रेम संख्या
- (b) बेस और लिमिट
- (c) पेज संख्या
- (d) प्राथमिकता

उत्तर: (b)



प्र9. पेज फ़ॉल्ट का अर्थ है:

- (a) प्रोग्राम में बग है
- (b) आवश्यक पेज मेमोरी में नहीं है और लाया जाना चाहिए
- (c) डिस्क विफल हो गई
- (d) प्रक्रिया हमेशा समाप्त होती है

उत्तर: (b)

प्र10. Optimal पेज प्रतिस्थापन उस पेज को निकालता है:

- (a) जो पहले लोड हुआ
- (b) जो सबसे कम ताज़ा उपयोग हुआ
- (c) जिसका अगला उपयोग भविष्य में सबसे दूर है
- (d) जिसका नंबर सबसे छोटा है

उत्तर: (c)

प्र11. Belady की विसंगति हो सकती है:

- (a) LRU में
- (b) Optimal में
- (c) FIFO में
- (d) सभी एल्गोरिदम में समान रूप से

उत्तर: (c)

प्र12. वर्चुअल मेमोरी आधारित है:

- (a) सभी पेज RAM में रखने पर
- (b) डिमांड पेजिंग पर — पेज केवल फ़ॉल्ट होने पर लोड करना
- (c) टेप स्टोरेज पर
- (d) केवल बड़ी डिस्क पर

उत्तर: (b)

प्र13. थ्रैशिंग का कारण है:

- (a) बहुत कम प्रक्रियाएँ
- (b) संयुक्त कार्यशील समूह उपलब्ध मेमोरी से अधिक होना
- (c) तेज़ डिस्क
- (d) हमेशा बड़ा पेज आकार

उत्तर: (b)

प्र14. Coalescing (सम्मिलन) का अर्थ है:

- (a) छिद्र को बाँटना
- (b) सटे मुक्त ब्लॉक को मिलाना
- (c) डिस्क संकुचित करना
- (d) swap out करना

उत्तर: (b)

प्र15. TLB है:

- (a) एक डिस्क
- (b) हाल के पेज-टेबल अनुवादों का तेज़ कैश
- (c) एक प्रकार का सेगमेंट
- (d) स्वैप फाइल

उत्तर: (b)

खंड B — लघु उत्तर प्रश्न (मॉडल उत्तर सहित)

प्र1. आधार/लिमिट उदाहरण के साथ तार्किक बनाम भौतिक एड्रेस। तार्किक एड्रेस वह एड्रेस है जैसा प्रोग्राम उसे देखता है (0 से उसका अपना स्पेस); भौतिक एड्रेस RAM में असली स्थान है। MMU हर तार्किक एड्रेस में प्रक्रिया का बेस जोड़ता है, लिमिट से जाँचने के बाद — बेस 12000, तार्किक 2500 → भौतिक 14500; लिमिट 4000 के साथ तार्किक 4100 एक सुरक्षा फ़ॉल्ट है।

प्र2. परिवर्तनशील विभाजन छिद्र क्यों बनाता है जबकि निश्चित विभाजन नहीं? परिवर्तनशील विभाजन ठीक आकार में काटे जाते हैं और मनमाने क्रम में मुक्त होते हैं, इसलिए मुक्त स्थान जीवित प्रक्रियाओं के बीच बिखरे ब्लॉक के रूप में निकलता है। निश्चित विभाजन पहले से कटे होते हैं और पूरे मुक्त होते हैं — वहाँ बर्बादी विभाजनों के अंदर (आंतरिक) रहती है, उनके बीच नहीं।

प्र3. पेज टेबल क्या है? एक प्रविष्टि क्या उत्तर देती है? प्रति प्रक्रिया एक तालिका जो पेज संख्या → फ्रेम संख्या का नक्शा बनाती है (वैध/अमान्य चिह्न सहित)। एक प्रविष्टि उत्तर देती है: "यदि यह पेज मेमोरी में है, तो किस फ्रेम में; यदि नहीं, तो फ़ॉल्ट उसे लाएगा।"

प्र4. सेगमेंटेशन बनाम पेजिंग — विभाजन आधार और विखंडन, दो लाइन। पेजिंग अंधाधुंध निश्चित आकारों में बाँटती है (कोई बाहरी विखंडन नहीं, छोटी आंतरिक बर्बादी)। सेगमेंटेशन तर्क द्वारा परिवर्तनशील आकारों में बाँटता है (स्वाभाविक अर्थ और सुरक्षा, लेकिन बाहरी विखंडन लौटता है)।

प्र5. पेज फ़ॉल्ट के छह चरण संक्षेप में। OS के पास ट्रेप → वैधता जाँच → फ्रेम खोजना/मुक्त करना (पीड़ित निकालना, गंदा हो तो वापस लिखना) → पेज पढ़कर लाना → पेज टेबल अद्यतन करना → फ़ॉल्ट करने वाले निर्देश को फिर शुरू करना।

प्र6. Optimal को पाठ्यपुस्तकों में क्यों रखा जाता है यदि वह कभी बनाया नहीं जा सकता? यह सिद्ध न्यूनतम-फ़ॉल्ट मापदंड है: असली रिकॉर्ड की गई संदर्भ स्ट्रिंगों पर इसे चलाना मापता है कि एक व्यावहारिक एल्गोरिदम (LRU) पूर्णता के कितने निकट आता है।

प्र7. स्थानीयता (locality) क्या है, और वर्चुअल मेमोरी इस पर क्यों निर्भर है? किसी छोटी समय खिड़की में प्रोग्राम पेजों का एक छोटा स्थिर समूह ही छूता है (लूप, गर्म कोड, निकट डेटा)। डिमांड पेजिंग केवल तभी सस्ती है जब फ़ॉल्ट दुर्लभ रहें — जो अच्छे व्यवहार वाले प्रोग्रामों के लिए स्थानीयता गारंटी करती है।

प्र8. संकुचन (Compaction) — क्या, कब, और इसकी लागत। बिखरे छिद्रों को एक ब्लॉक में मिलाने के लिए सभी प्रक्रियाओं को साथ खिसकाना ताकि बड़ी माँग समा सके। तब उपयोग होता है जब कुल मुक्त मेमोरी पर्याप्त हो लेकिन कोई छिद्र न हो; लागत नकल समय और सिस्टम ठहराव है, और सभी एड्रेस फिर से आधारित करने पड़ते हैं।

खंड C — दीर्घ उत्तर प्रश्न (मॉडल उत्तर सहित)

प्र1. तीन स्थान रणनीतियों की व्याख्या कीजिए। छिद्र 100/500/200/300/600 पर माँगों 212/417/112/426 के साथ, दिखाइए कि केवल एक रणनीति चारों को रखती है, और सामान्य सबक निकालिए। मुख्य बिंदु: परिभाषाएँ (पहला पर्याप्त; सबसे छोटा पर्याप्त; सबसे बड़ा); अध्याय 7 §7.3 के पूरे हल (959 KB मुक्त के बीच First Fit 426 पर विफल; Worst Fit उसी प्रकार विफल; Best Fit 600 को 426 के लिए बचाकर सबको रखती है); सबक — स्थान कल की माँगों के बारे में एक लंबा खेल है; व्यवहार में First Fit $\approx$ Best Fit समग्र रूप से, Worst Fit सबसे कमज़ोर, और Best Fit की झिल्लियाँ नोट की गई; बाहरी विखंडन और संकुचन से जोड़िए।

प्र2. पेजिंग में एड्रेस अनुवाद का चित्र वर्णन और हल उदाहरण सहित वर्णन कीजिए। बाहरी विखंडन क्यों नहीं होता, और अंतिम पेज क्या बर्बाद करता है? मुख्य बिंदु: एड्रेस = $p + d$; पेज टेबल $p \rightarrow f$; भौतिक = f, d अपरिवर्तित (चित्र 8.1 वर्णन); हल उदाहरण (पेज आकार 100, तार्किक 753 $\rightarrow$ भौतिक 453); फ्रेम विनिमय हैं इसलिए कोई भी मुक्त फ्रेम किसी भी पेज की सेवा करता है — छिद्र बन ही नहीं सकते; अंतिम पेज की अनुपयुक्त पूँछ आंतरिक विखंडन है, प्रति प्रक्रिया अधिकतम एक पेज; पेज-टेबल आकार और TLB का उल्लेख।

प्र3. आंतरिक और बाहरी विखंडन की तुलना कीजिए: परिभाषाएँ, हर एक कहाँ होता है (योजनाओं के नाम), और उपचार। उद्योग पेजिंग पर क्यों स्थिर हुआ? मुख्य बिंदु: आंतरिक = आवंटित ब्लॉक के अंदर (निश्चित विभाजन, अंतिम पेज); बाहरी = बिखरे मुक्त छिद्र (परिवर्तनशील विभाजन, सेगमेंटेशन) 426-का-959-KB दृष्टांत सहित; उपचार — आंतरिक: छोटी इकाइयों/ब्लॉक का व्यापार-बंद; बाहरी: सम्मिलन (आंशिक), संकुचन (महंगा), या सततता को ही हटाना; पेजिंग सततता आवश्यकता हटाती है, इसलिए कोई बाहरी विखंडन नहीं — निर्णायक लाभ, पेज टेबल और छोटी आंतरिक बर्बादी की कीमत पर।

प्र4. वर्चुअल मेमोरी क्या है? डिमांड पेजिंग, उसके लाभ, स्थानीयता धारणा, और थ्रैशिंग उसके दो उपचारों सहित समझाइए। मुख्य बिंदु: RAM से बड़ा प्रति-प्रक्रिया वर्चुअल स्पेस, डिस्क पर पेज फ़ॉल्ट पर लाए गए; लाभ (RAM से बड़े प्रोग्राम, अधिक निवासी प्रक्रियाएँ, कम लोड I/O); भार-वहन धारणा के रूप में स्थानीयता; थ्रैशिंग तंत्र — संयुक्त कार्यशील समूह > RAM → फ़ॉल्ट तूफ़ान → CPU पतन → शेड्यूलर और प्रवेश देता है (घातक); उपचार: कार्यशील-समूह फ्रेम की गारंटी या मल्टीप्रोग्रामिंग की मात्रा घटाना; CPU-उपयोग वक्र वर्णित (वृद्धि, चरम, चट्टान)।

खंड D — हल किए गए आंकिक प्रश्न

N1. (अध्याय 7 अभ्यास सेट, हल किया गया।) क्रम में छिद्र: 200, 600, 250, 450, 100। माँगें: 380, 120, 290, 500।

- **First Fit:** 380 → 600 (पहला जो बैठता है), शेष 220। 120 → 200, शेष 80। 290 → 450, शेष 160। 500 → छिद्र हैं 80, 220, 250, 160, 100 — **रखा नहीं जा सकता।**
- **Best Fit:** 380 → 450, शेष 70। 120 → 200, शेष 80। 290 → 600, शेष 310। 500 → सबसे बड़ा छिद्र 310 है — **रखा नहीं जा सकता।**
- **Worst Fit:** 380 → 600, शेष 220। 120 → 450, शेष 330। 290 → 330, शेष 40। 500 → छिद्र 200, 220, 250, 40, 100 — **रखा नहीं जा सकता।**

अंक लायक सबक: तीनों 500 माँग पर विफल होते हैं हालाँकि कुल 810 KB मुक्त खड़ा है — सबसे शुद्ध प्रदर्शन कि बाहरी विखंडन टुकड़ों के बारे में है, मात्रा के बारे में नहीं; केवल संकुचन (या पेजिंग) इस मेमोरी को बचाता है।

N2. (अध्याय 8 अभ्यास सेट, हल किया गया।) (a) पेज आकार 200, पेज टेबल पेज 5 → फ्रेम 11, तार्किक 1130 का अनुवाद: $p = 1130 \div 200 = 5$, $d = 1130 - 5 \times 200 = 130$; भौतिक = $11 \times 200 + 130 = 2330$ ।

(b) उसी अध्याय की सेगमेंटेशन जाँच: सेगमेंट 2, बेस 5000, लिमिट 800, ऑफ़सेट 100 → $100 < 800$

✓ → भौतिक = **5100** (ऑफ़सेट 900 फ़ॉल्ट करता — लिमिट ही सुरक्षा है)।

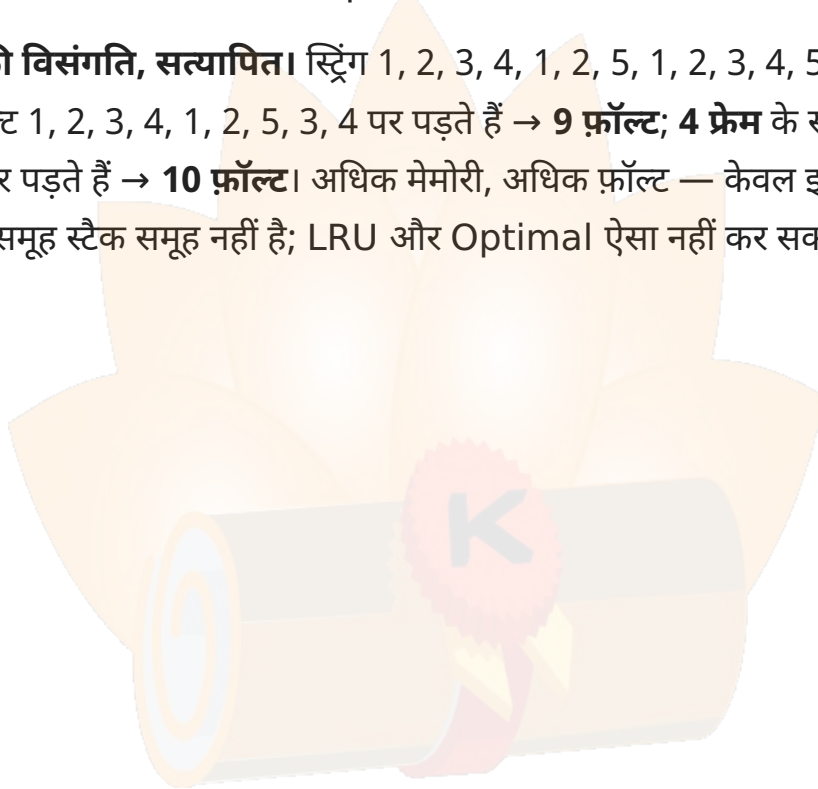
(c) 8 KB पेजों वाला 32-बिट सिस्टम: ऑफ़सेट बिट = $\log_2(8192) = 13$; पेज-संख्या बिट = $32 - 13 = 19$ → पेज टेबल को $2^{19} = 524,288$ प्रविष्टियाँ चाहिए हो सकती हैं।

N3. (अध्याय 9 अभ्यास सेट, हल किया गया।) संदर्भ स्ट्रिंग **1, 2, 3, 2, 1, 4, 3, 3** फ्रेम के साथ।

- **FIFO:** 1 F, 2 F, 3 F, 2 hit, 1 hit, 4 F (1 को निकालता है), 3 hit → **4 फ़ॉल्ट**।
- **LRU:** 1 F, 2 F, 3 F, 2 hit, 1 hit, 4 F (3 को निकालता है, सबसे कम ताज़ा उपयोग), 3 F (2 को निकालता है) → **5 फ़ॉल्ट**।
- **Optimal:** 1 F, 2 F, 3 F, 2 hit, 1 hit, 4 F (उस पेज को निकालता है जो फिर कभी उपयोग नहीं होगा — 1 या 2), 3 hit → **4 फ़ॉल्ट**।

एक स्ट्रिंग जहाँ LRU FIFO से हारता है — दोनों सबक रखिए: LRU आमतौर पर जीतता है, कोई अनुभवजन्य नियम हमेशा नहीं जीतता, Optimal सबकी सीमा बाँधता है।

N4. Belady की विसंगति, सत्यापित। स्ट्रिंग 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 FIFO के अंतर्गत: **3 फ्रेम** के साथ फ़ॉल्ट 1, 2, 3, 4, 1, 2, 5, 3, 4 पर पड़ते हैं → **9 फ़ॉल्ट**; **4 फ्रेम** के साथ वे 1, 2, 3, 4, 5, 1, 2, 3, 4, 5 पर पड़ते हैं → **10 फ़ॉल्ट**। अधिक मेमोरी, अधिक फ़ॉल्ट — केवल इसलिए संभव क्योंकि FIFO का निवासी समूह स्टैक समूह नहीं है; LRU और Optimal ऐसा नहीं कर सकते।





प्रश्न बैंक — इकाई 4: I/O सिस्टम

(BTEUP CSE तीसरा सेमेस्टर · विषय 3.1 · पेपर में इकाई का भार: लगभग 18%)

खंड A — वस्तुनिष्ठ प्रश्न (MCQ)

प्र1. मास स्टोरेज की कौन सी विशेषता RAM में नहीं है?

- (a) गति
- (b) गैर-अस्थिरता
- (c) CPU द्वारा बाइट एड्रेसिंग
- (d) छोटा आकार

उत्तर: (b)

प्र2. डिस्क प्लैटर पर सबसे छोटी भौतिक स्टोरेज इकाई है:

- (a) ट्रैक
- (b) सिलेंडर
- (c) सेक्टर
- (d) ब्लॉक

उत्तर: (c)

प्र3. सिलेंडर है:

- (a) एक ट्रैक
- (b) सभी सतहों पर एक ही भुजा स्थिति पर ट्रैक
- (c) एक सेक्टर समूह
- (d) डिस्क का कैश

उत्तर: (b)

प्र4. 6000 RPM पर औसत घूर्णन विलंब है:

- (a) 10 ms
- (b) 2.5 ms
- (c) 5 ms
- (d) 20 ms

उत्तर: (c) — एक घूर्णन 10 ms लेता है; औसत प्रतीक्षा आधा घूर्णन है।

प्र5. NAS नेटवर्क पर ____ निर्यात करता है; SAN ____ निर्यात करता है।

- (a) ब्लॉक; फाइलें
- (b) फाइलें; ब्लॉक
- (c) सेक्टर; फाइलें
- (d) फाइलें; सेक्टर

उत्तर: (b)

प्र6. डिस्क शेड्यूलिंग मुख्य रूप से घटाती है:

- (a) स्थानांतरण समय
- (b) घूर्णन विलंब
- (c) सीक समय / कुल हेड गति
- (d) डिस्क आकार

उत्तर: (c)

प्र7. कौन सा डिस्क एल्गोरिदम दूर की माँगों को भूखा रख सकता है?

- (a) FCFS
- (b) SSTF
- (c) SCAN
- (d) C-SCAN

उत्तर: (b)

प्र8. C-SCAN SCAN से इस बात में भिन्न है कि वह:

- (a) दोनों दिशाओं में सेवा करता है
- (b) एक दिशा में सेवा करता है और शुरुआत पर लौटता है
- (c) कोई कतार उपयोग नहीं करता
- (d) केवल SSD के लिए है

उत्तर: (b)

प्र9. स्वैप स्थान प्रबंधित होता है:

- (a) साझा करने के लिए साधारण फाइलों के रूप में
- (b) OS द्वारा सीधे, फाइल सिस्टम को बायपास करके
- (c) उपयोगकर्ता द्वारा
- (d) टेप पर

उत्तर: (b)

प्र10. RAID 0 देता है:

- (a) मिररिंग
- (b) पैरिटी
- (c) केवल striping, कोई अतिरिक्त नहीं
- (d) दो-डिस्क विफलता सुरक्षा

उत्तर: (c)

प्र11. 5-डिस्क RAID 5 सरणी एक साथ कितनी डिस्क विफलताएँ सहती है?

- (a) 0
- (b) 1
- (c) 2
- (d) 5

उत्तर: (b)

प्र12. n-डिस्क RAID 6 सरणी में उपयोगी स्थान है:

- (a) n डिस्क के बराबर
- (b) $(n - 1)$
- (c) $(n - 2)$
- (d) $n/2$

उत्तर: (c)

प्र13. RAID 5 में पैरिटी है:

- (a) हर डिस्क की पूर्ण प्रतिलिपि
- (b) डिस्क में वितरित रखी गई, XOR द्वारा गिनी गई
- (c) टेप पर रखी गई
- (d) बूट ब्लॉक

उत्तर: (b)

प्र14. विभाजन की लॉजिकल फॉर्मेटिंग लिखती है:

- (a) सेक्टर
- (b) एक फाइल सिस्टम
- (c) BIOS
- (d) पैरिटी

उत्तर: (b)

प्र15. डिस्क शेड्यूलिंग सबसे कम मायने रखती है:

- (a) भार के अंतर्गत HDD के लिए
- (b) खाली HDD के लिए
- (c) SSD/NVMe के लिए
- (d) HDD वाले व्यस्त फाइल सर्वर के लिए

उत्तर: (c) — कोई भुजा नहीं, कोई सीक नहीं।

खंड B — लघु उत्तर प्रश्न (मॉडल उत्तर सहित)

प्र1. सेक्टर, ट्रैक, सिलेंडर — एक-एक लाइन। सेक्टर ट्रैक की सबसे छोटी भौतिक इकाई है (पारंपरिक रूप से 512 B)। ट्रैक किसी सतह पर सेक्टरों का एक संकेंद्रित वलय है। सिलेंडर सभी सतहों पर एक भुजा स्थिति पर ट्रैक का समूह है — सब भुजा को हिलाए बिना पहुँच योग्य।

प्र2. घूर्णन विलंब औसत के रूप में क्यों बताया जाता है? जब कोई माँग आती है, तो वांछित सेक्टर अपने घूर्णन में कहीं भी हो सकता है — हेड के ठीक नीचे से गुजरता हुआ (लगभग 0 प्रतीक्षा) या ठीक छूटा हुआ (पूरा घूर्णन)। यादृच्छिक स्थितियों पर अपेक्षित मान आधा घूर्णन है।

प्र3. होस्ट-अटैच्ड स्टोरेज बनाम NAS, दो बिंदु। होस्ट-अटैच्ड: डिस्क एक कंप्यूटर की बस से तारबद्ध (SATA/NVMe), केवल उसी द्वारा देखी गई। NAS: LAN पर एक उपकरण जो कई क्लाइंट को फाइलें देता है, जो कभी कच्चे ब्लॉक नहीं देखते।

प्र4. डिस्क के लिए FCFS बनाम SSTF, दो बिंदु। FCFS आगमन क्रम में सेवा करता है — निष्पक्ष, लेकिन भुजा ज़िगज़ैग करती है (अध्याय की कतार में 640)। SSTF सबसे निकट प्रतीक्षारत माँग की सेवा करता है — कहीं कम यात्रा (उसी कतार पर 236) लेकिन स्थिर मध्य भार के अंतर्गत किनारे भूखे रह सकते हैं।

प्र5. SCAN में लिफ्ट विचार क्या है? यात्रा की दिशा में सब कुछ सेवित कीजिए, केवल अंत में उलटिए (या LOOK में, अंतिम माँग पर) — कोई माँग एक पूरे स्वीप से अधिक प्रतीक्षा नहीं करती, इसलिए यात्रा बचत भुखमरी नहीं खरीदती।

प्र6. स्वैप विभाजन बनाम स्वैप फाइल — एक-एक लाइन। विभाजन: एक कच्चा समर्पित डिस्क क्षेत्र — सबसे तेज़, लेकिन आकार निश्चित। फाइल: सामान्य फाइल सिस्टम के अंदर एक फाइल — बनाना/आकार बदलना आसान, थोड़ी धीमी।

प्र7. RAID 1 की कुशलता ठीक 50% क्यों है? हर डेटा ब्लॉक दो बार लिखा जाता है, दर्पण युग्म के हर एक पर; $2n$ डिस्क की क्षमता में से, n डिस्क के बराबर प्रतियाँ हैं, इसलिए कच्चे स्थान का आधा उपयोगी है।

प्र8. "RAID बैकअप नहीं है" — दो लाइनों में समझाइए। RAID केवल डिस्क हार्डवेयर विफलता से बचाता है। विलोपन, भ्रष्टाचार, रैंसमवेयर, चोरी और आग सभी प्रतियों पर एक साथ प्रहार करते हैं — बैकअप एक अलग प्रतिलिपि है, आदर्श रूप से कहीं और और ऑफ़लाइन।

खंड C — दीर्घ उत्तर प्रश्न (मॉडल उत्तर सहित)

प्र1. हार्ड डिस्क की संरचना का नामांकित वर्णन कीजिए (प्लैटर, हेड, ट्रैक, सेक्टर, सिलेंडर) और 8 सतहों, प्रति सतह 1024 ट्रैक, प्रति ट्रैक 64 सेक्टर और प्रति सेक्टर 512 बाइट वाली डिस्क की क्षमता गिनिए। मुख्य बिंदु: रचना (लेपित प्लैटर, साथ चलती भुजा पर हेड, संकेद्रित ट्रैक, सबसे छोटी इकाई के रूप में सेक्टर, सिलेंडर = एक ही स्थिति के ट्रैक); क्षमता = $8 \times 1024 \times 64 \times 512 = 268,435,456 \text{ B} = 256 \text{ MB}$; नोट कीजिए कि आधुनिक डिस्क इस ज्यामिति को लॉजिकल ब्लॉक क्रमांकन के पीछे छिपाती हैं।

प्र2. डिस्क एक्सेस समय के तीन घटक परिभाषित कीजिए। शेड्यूलिंग किस पर हमला करती है, और अन्य दो पर क्यों नहीं कर सकती? मुख्य बिंदु: सीक (ट्रैक तक भुजा — माँग क्रम पर निर्भर, इसलिए शेड्यूलिंग); घूर्णन विलंब (प्लैटर घूर्णन — RPM और संयोग स्थिति द्वारा निर्धारित, क्रम द्वारा नहीं); स्थानांतरण (हेड से गुजरता डेटा — आकार और घूर्णन गति द्वारा निर्धारित)। निष्कर्ष: शेड्यूलिंग लीवर केवल सीक पर काम करता है, जो सौभाग्य से प्रमुख और सबसे परिवर्तनशील पद है।

प्र3. अध्याय की कतार (माँगें 98, 183, 37, 122, 14, 124, 65, 67; हेड 53) पर FCFS, SSTF, SCAN और C-SCAN की तुलना कीजिए। क्रम और योग दीजिए, और भुखमरी पर चर्चा कीजिए। मुख्य बिंदु: FCFS 640 (आगमन क्रम, कोई भुखमरी नहीं, ज़िगज़ैग); SSTF 236 (लालची सबसे निकट, प्रदर्शित करने वाले माँग प्रतिरूप सहित किनारों की भुखमरी); SCAN 331 (अंत 199 तक स्वीप, कोई भुखमरी नहीं); C-SCAN 382 वापसी कूद गिनते हुए (समान प्रतीक्षा; नियम बताया गया); सारांश तालिका और निष्पक्षता-बनाम-यात्रा समझौता।

प्र4. RAID स्तर 0, 1, 5, 6 और 10 की तकनीक, न्यूनतम डिस्क, स्थान कुशलता और विफलता सहनशीलता सहित व्याख्या कीजिए। एक छोटा व्यवसाय अपने खाता सर्वर की सुरक्षा करता है: RAID 5 और RAID 10 के बीच चुनिए और चुनाव का बचाव कीजिए। मुख्य बिंदु: अध्याय 11 की तालिका; पैरिटी और XOR पुनर्निर्माण वर्णित; चुनाव किसी भी प्रकार से कारणों सहित बचाव किया गया (RAID 5: अधिक उपयोगी स्थान, एक-डिस्क सहनशीलता, धीमा पुनर्निर्माण और लेखन दंड; RAID 10: आधा स्थान गया, तेज़ पुनर्निर्माण और लेखन, प्रति दर्पण युग्म एक सहता है) — व्यवसाय के डेटा मूल्य और बंद समय लागत से बँधा निष्कर्ष ही अंक दिलाता है, स्तर का अक्षर नहीं।

खंड D — हल किए गए आंकिक प्रश्न

N1. (अध्याय 11 अभ्यास सेट, हल किया गया।) कतार: 55, 120, 18, 88, 160, 42; हेड 40 पर; डिस्क 0-199।

- **FCFS:** 40 → 55 (15) → 120 (65) → 18 (102) → 88 (70) → 160 (72) → 42 (118)।
कुल = 442।
- **SSTF:** 40 से, सबसे निकट 42 है (2) → 55 (13) → 88 (33) → 120 (32) → 160 (40) → 18 (142)। कुल = 262।
- **SCAN (ऊपर बढ़ता हुआ, अंत तक):** 42, 55, 88, 120, 160, फिर अंत 199, वापस 18 तक।
गति 2 + 13 + 33 + 32 + 40 + 39 + 181 = 340।
- **C-LOOK (ऊपर बढ़ता हुआ):** क्रम 42, 55, 88, 120, 160, फिर सीधे 18 पर कूद: 2 + 13 + 33 + 32 + 40 + 142 = 262 — यहाँ SSTF के कुल से बराबरी, उसकी भुखमरी जोखिम के बिना; केवल योग पूरी कहानी कभी नहीं बताते।

N2. (अध्याय 10 अभ्यास सेट, हल किया गया।) डिस्क: 10 सतहें, प्रति सतह 4096 ट्रैक, प्रति ट्रैक 256 सेक्टर, प्रति सेक्टर 512 बाइट। प्रति सतह सेक्टर = $4096 \times 256 = 1,048,576$; प्रति सतह बाइट = $1,048,576 \times 512 = 512 \text{ MiB}$; कुल = $10 \times 512 \text{ MiB} = 5120 \text{ MiB} = 5 \text{ GiB}$ ।

N3. RAID अंकगणित। हर एक 3 TB की छह डिस्क:

- **RAID 5:** उपयोगी = $(6 - 1) \times 3 = 15 \text{ TB}$; 1 विफलता सहता है।
- **RAID 6:** उपयोगी = $(6 - 2) \times 3 = 12 \text{ TB}$; 2 विफलताएँ सहता है।
- **RAID 10:** उपयोगी = $6 \times 3 / 2 = 9 \text{ TB}$; प्रति दर्पण युग्म 1 विफलता सहता है (भाग्य उन्हें फैलाए तो 3 तक, गारंटी केवल 1 की)।
- **RAID 0:** उपयोगी = 18 TB; 0 सहता है — और पहली विफलता पर पूरी सरणी खोता है।



प्रश्न बैंक — इकाई 5: फाइल प्रबंधन

(BTEUP CSE तीसरा सेमेस्टर · विषय 3.1 · पेपर में इकाई का भार: लगभग 18%)

खंड A — वस्तुनिष्ठ प्रश्न (MCQ)

प्र1. OS आंतरिक रूप से फाइल को पहचानता है:

- (a) उसके नाम से
- (b) उसके अद्वितीय पहचानकर्ता/inode नंबर से
- (c) उसके एक्सटेंशन से
- (d) उसके आइकन से

उत्तर: (b)

प्र2. कौन सा मानक फाइल कार्य नहीं है?

- (a) बनाना
- (b) reposition
- (c) compile
- (d) truncate

उत्तर: (c)

प्र3. टेप स्टोरेज किस एक्सेस विधि का समर्थन करता है?

- (a) प्रत्यक्ष
- (b) अनुक्रमित
- (c) केवल अनुक्रमिक
- (d) हैश की गई

उत्तर: (c)

प्र4. रिकॉर्ड 5000 को सीधे लाने वाला डेटाबेस उपयोग करता है:

- (a) अनुक्रमिक एक्सेस
- (b) प्रत्यक्ष एक्सेस
- (c) टेप एक्सेस
- (d) केवल append

उत्तर: (b)

प्र5. द्वि-स्तरीय डायरेक्टरी में, नाम टकराव टलते हैं:

- (a) सभी उपयोगकर्ताओं में
- (b) केवल अलग उपयोगकर्ताओं की डायरेक्टरी के बीच
- (c) कभी नहीं
- (d) लंबे नामों से

उत्तर: (b)

प्र6. पथ /home/student/notes.txt है:

- (a) सापेक्ष
- (b) निरपेक्ष
- (c) एक लिंक
- (d) एक माउंट बिंदु

उत्तर: (b)

प्र7. माउंटिंग एक फाइल सिस्टम को जोड़ता है:

- (a) CPU से
- (b) मौजूद डायरेक्टरी वृक्ष में एक माउंट बिंदु पर
- (c) केवल नेटवर्क से
- (d) एक फाइल के अंदर

उत्तर: (b)

प्र8. UNIX अनुमतियों `-rwxr-x---` में, समूह वर्ग कर सकता है:

- (a) पढ़ना और लिखना
- (b) पढ़ना और निष्पादित करना
- (c) केवल निष्पादित करना
- (d) कुछ नहीं

उत्तर: (b)

प्र9. ACL, rwx वर्गों की तुलना में, देता है:

- (a) कम सटीकता
- (b) लंबी सूचियों की लागत पर प्रति-उपयोगकर्ता सटीकता
- (c) कोई सुरक्षा नहीं
- (d) हमेशा तेज़ जाँच

उत्तर: (b)

प्र10. UNIX में FCB को कहते हैं:

- (a) inode
- (b) FAT
- (c) PCB
- (d) TLB

उत्तर: (a)

प्र11. कौन सी आवंटन विधि केवल प्रारंभ ब्लॉक + लंबाई रखती है?

- (a) शृंखलाबद्ध
- (b) अनुक्रमित
- (c) सतत
- (d) हैश की गई

उत्तर: (c)

प्र12. शृंखलाबद्ध आवंटन की सबसे बुरी कमज़ोरी है:

- (a) बाहरी विखंडन
- (b) बहुत खराब प्रत्यक्ष एक्सेस
- (c) फाइल वृद्धि
- (d) कोई अनुक्रमिक एक्सेस नहीं

उत्तर: (b)

प्र13. FAT शृंखलाबद्ध आवंटन को सुधारता है:

- (a) सूचक हटाकर
- (b) सभी सूचकों को एक तालिका में एकत्र करके जिसे कैश किया जा सकता है
- (c) ब्लॉक को सतत बनाकर
- (d) पैरिटी जोड़कर

उत्तर: (b)

प्र14. मुक्त स्थान प्रबंधन के लिए बिट वेक्टर उपयोग करता है:

- (a) प्रति ब्लॉक एक बाइट
- (b) प्रति ब्लॉक एक बिट
- (c) प्रति फाइल एक ब्लॉक
- (d) पैरिटी बिट

उत्तर: (b)

प्र15. FAT32 पर अधिकतम फाइल आकार है:

- (a) 2 GB
- (b) 4 GB
- (c) 16 GB
- (d) असीमित

उत्तर: (b)

खंड B — लघु उत्तर प्रश्न (मॉडल उत्तर सहित)

प्र1. नाम बनाम पहचानकर्ता — OS दोनों क्यों रखता है? नाम मनुष्यों की सेवा करता है (बदला जा सकता है, प्रति डायरेक्टरी, वैश्विक रूप से अद्वितीय नहीं)। पहचानकर्ता (inode नंबर) फाइल के रिकॉर्ड के लिए OS की अपनी स्थिर कुंजी है, हर आंतरिक संरचना द्वारा उपयोग की गई; नाम डायरेक्टरी के द्वारा पहचानकर्ताओं में सुलझते हैं।

प्र2. open() और close() पर क्या होता है? Open फाइल के FCB को सिस्टम खुली-फाइल तालिका में लोड करता है, शुरुआत में अपने सूचक के साथ प्रति-प्रक्रिया प्रविष्टि बनाता है, और एक हैंडल लौटाता है। Close प्रति-प्रक्रिया प्रविष्टि हटाता है; जब अंतिम बंद करने वाला जाता है, तो FCB प्रतिलिपि डिस्क पर लौटती है, बफ़र किए गए लेखन फ्लश सहित।

प्र3. अनुक्रमिक एक्सेस — सूचक का व्यवहार और दो स्वाभाविक कार्यभार। सूचक शुरुआत से शुरू होता है और हर पठन/लेखन के साथ स्वतः आगे बढ़ता है; कोई कूद नहीं होती। CCTV फुटेज रिकॉर्डिंग और लॉग फाइल को ऊपर से नीचे तक पढ़ने के लिए स्वाभाविक।

प्र4. सापेक्ष पथ — .. का उपयोग करके एक उदाहरण सहित परिभाषित कीजिए। जड़ के बजाय वर्तमान कार्यशील डायरेक्टरी से व्याख्या किया गया पथ। /home/student से, फाइल /home/teacher/marks/cs.xlsx, ../teacher/marks/cs.xlsx है — .. पहले /home तक चढ़ता है।

प्र5. सत्र अर्थविज्ञान (session semantics) एक छोटे अनुच्छेद में। अधिकांश नेटवर्क फाइल सिस्टम द्वारा उपयोग किए गए मॉडल के अंतर्गत, किसी खुली फाइल में उपयोगकर्ता द्वारा किए गए बदलाव दूसरों को तब दिखाई देते हैं जब फाइल बंद होती है — सत्र का काम संपादन के दौरान बाइट दर बाइट नहीं, अंत में पूरे के रूप में प्रकट होता है।

प्र6. डायरेक्टरी अनुमति सूक्ष्मता: पठनीय फाइल, अपहुँच फाइल। उस डायरेक्टरी के अंदर पढ़ने की अनुमति वाली फाइल, जिसके लिए आपके पास execute (खोज) अनुमति नहीं है, खोली नहीं जा सकती — OS उसे पाने के लिए पथ पार ही नहीं कर सकता। डायरेक्टरी अधिकार सड़क की रक्षा करते हैं, फाइल अधिकार गंतव्य की।

प्र7. समूहीकरण (मुक्त स्थान) दो लाइनों में। पहला मुक्त ब्लॉक मुक्त ब्लॉक के एक समूह के पते रखता है; समूह का अंतिम पता अगले ऐसे इंडेक्स ब्लॉक की ओर इशारा करता है। लंबी मुक्त सूचियाँ इस प्रकार एक बार में एक टुकड़ा लाई जाती हैं, शृंखलाबद्ध सूची के ब्लॉक-दर-ब्लॉक रेंगने से बचते हुए।

प्र8. जर्नलिंग — यह किस समस्या को हल करता है? अद्यतन के बीच में क्रैश फाइल सिस्टम को आधा-लिखा और असंगत छोड़ सकता है। जर्नलिंग सिस्टम इच्छित बदलावों को पहले एक लॉग में लिखता है; क्रैश के बाद वह लॉग प्रविष्टि को फिर चलाता या त्यागता है, वॉल्यूम को शीघ्र संगत अवस्था में लौटाते हुए।

खंड C — दीर्घ उत्तर प्रश्न (मॉडल उत्तर सहित)

प्र1. फाइल गुणों और कार्यों को गिनाइए। फिर अनुक्रमिक और प्रत्यक्ष एक्सेस की एक-एक कार्यभार के साथ तुलना कीजिए, यह समझाते हुए कि उनमें से एक के लिए टेप प्रश्न का फैसला क्यों करता है। मुख्य बिंदु: गुण (नाम, पहचानकर्ता, प्रकार, स्थान, आकार, सुरक्षा, समय-मुहरें, स्वामी); कार्य (बनाना, लिखना, पढ़ना, reposition, मिटाना, truncate, open/close); अनुक्रमिक (सूचक क्रम, टेप/लॉग/मीडिया) बनाम प्रत्यक्ष (ब्लॉक-नंबर स्थिति निर्धारण, डिस्क पर डेटाबेस) स्थिति सूत्र सहित (सापेक्ष ब्लॉक × रिकॉर्ड आकार); टेप केवल अनुक्रमिक का समर्थन करता है — टेप पर डेटाबेस इसलिए व्यवहार में असंभव है, माध्यम कितना भी सस्ता क्यों न हो।

प्र2. डायरेक्टरी संरचनाओं को एकल-स्तरीय से अचक्रीय ग्राफ़ तक खोजिए: संरचना, हर चरण ने जो समस्या हल की, और हर एक ने जो नई समस्या बनाई। मुख्य बिंदु: एकल-स्तरीय (सभी फाइलें साथ — टकराव और अव्यवस्था); द्वि-स्तरीय (प्रति-उपयोगकर्ता स्थान — टकराव सुलझे, कोई आंतरिक संगठन या साझा करना नहीं); वृक्ष (असीमित घोंसला, निरपेक्ष/सापेक्ष पथ, कार्यशील डायरेक्टरी — विलोपन नीति प्रश्न); अचक्रीय ग्राफ़ (लिंक और साझा फाइलें — एक ही फाइल दो नामों के नीचे; विलोपन को संदर्भ गिनती चाहिए; खोजों को सीमित रखने के लिए घेरे फिर भी वर्जित)।

प्र3. सतत, शृंखलाबद्ध और अनुक्रमित आवंटन की तुलना कीजिए: रिकॉर्ड सामग्री, अनुक्रमिक एक्सेस, प्रत्यक्ष एक्सेस, बाहरी विखंडन, वृद्धि, और अतिरिक्त लागत पर। (i) CCTV वीडियो अभिलेख (ii) छात्र-रिकॉर्ड डेटाबेस के लिए चुनिए, दोनों का बचाव करते हुए। मुख्य बिंदु: अध्याय 13 की तुलना तालिका गद्य में; शृंखलाबद्ध आवंटन की FAT मरम्मत समझाई गई; चुनाव — CCTV: सतत (एक बार लिखा गया, अनुक्रमिक रूप से देखा गया, वृद्धि की खराब योजना इसका एक जोखिम है — या दर्दरहित वृद्धि के लिए शृंखलाबद्ध; बिना तर्क के चुनाव से बचाव किया गया कोई भी बेहतर); डेटाबेस: अनुक्रमित (प्रत्यक्ष एक्सेस ही कार्यभार है)।

प्र4. सुरक्षा समझाइए: छह एक्सेस प्रकार, ACL बनाम स्वामी/समूह/अन्य योजना (सटीकता बनाम संकुचितता), और `-rwxr-x---` को पूरी तरह डिकोड कीजिए। डायरेक्टरी को भी सुरक्षित क्यों किया जाना चाहिए? मुख्य बिंदु: read/write/execute/append/delete/list; ACL (प्रति-उपयोगकर्ता सूचियाँ, सटीक, भारी) बनाम rwx (9 बिट, तीन वर्ग, एक अतिरिक्त उपयोगकर्ता का नाम नहीं ले सकता); डिकोड: स्वामी rwx (सब कुछ), समूह r-x (पढ़ना+निष्पादित), अन्य --- (कुछ नहीं), अग्रणी '-' एक साधारण फाइल को चिह्नित करता है; डायरेक्टरी: execute/खोज अनुमति खुद पथ का द्वार है (B6), और list अनुमति नामों को देखना नियंत्रित करती है।

खंड D — हल किए गए आंकिक और अभ्यास

N1. (अध्याय 13 अभ्यास सेट, हल किया गया।) डिस्क पर मुक्त ब्लॉक: {2, 3, 4, 9, 15, 16, 17, 30}।

- **शृंखलाबद्ध सूची:** मुक्त-सूची का सिरा 2 की ओर इशारा करता है, जिसकी पूँछ 3 रखती है, फिर 4, फिर 9, फिर 15, 16, 17, और 30 शृंखला समाप्त करता है (null)। आठ ब्लॉक गिनने के लिए आठ सूचक यात्रा किए गए — योजना की ज्ञात धीमी गति।
- **गिनती (Counting):** धाराएँ (पहला, गिनती) जोड़ों के रूप में वर्णित: (2, 3), (9, 1), (15, 3), (30, 1) — चार छोटी प्रविष्टियाँ उसी स्थान का ठीक वर्णन करती हैं, क्योंकि मुक्त स्थान धाराओं में आता है।

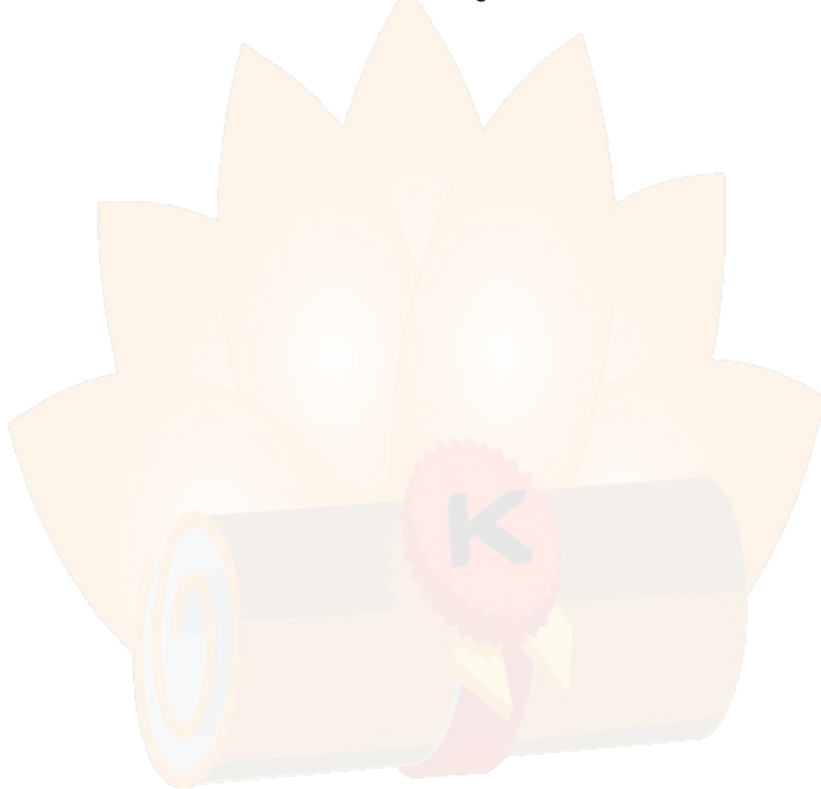
N2. बिटमैप आकार (हल किया गया)। 4 KB ब्लॉक (2^{12} बाइट) वाली 2 TB डिस्क (2^{41} बाइट): ब्लॉक की संख्या = $2^{41} \div 2^{12} = 2^{29}$ । हर एक के लिए एक बिट $\rightarrow 2^{29}$ बिट = 2^{26} बाइट = **64 MiB** का बिटमैप — केवल ज्यामिति द्वारा निश्चित, डिस्क खाली हो या लगभग भरी, अपरिवर्तित।

N3. पथ (अध्याय 12 अभ्यास सेट, हल किया गया)। निरपेक्ष पथ `/home/teacher/marks/cs.xlsx`; कार्यशील डायरेक्टरी `/home/student`। सापेक्ष पथ: `../teacher/marks/cs.xlsx` (student से home तक चढ़िए, teacher/marks में उतरिए)। और उलटा कौशल: `/home/student` से, सापेक्ष पथ `notes/os.txt` निरपेक्ष `/home/student/notes/os.txt` को नामित करता है।

N4. अनुमति निर्णय। (a) `-rw-r-----`: स्वामी पढ़ता+लिखता है; समूह केवल पढ़ता है; अन्य कुछ नहीं।

(b) एक शिक्षिका चाहती है: वह अपनी अंक फाइल पढ़-लिख सके, उसका विभाग समूह उसे पढ़ सके, कोई अन्य कुछ नहीं, और फाइल किसी के द्वारा निष्पादित न हो। आवश्यक स्ट्रिंग: `-rw-r-----` — ठीक (a)।

(c) एक साझा स्क्रिप्ट जिसे स्वामी संपादित करता है और सब (स्वामी, समूह, अन्य) चलाते हैं लेकिन कोई अन्य संपादित नहीं करता: स्वामी को `rw` (7) चाहिए, समूह `r-x` (5), अन्य `r-x` (5) → `-rwxr-xr-x`।





मॉडल पेपर — ऑपरेटिंग सिस्टम (विषय 3.1)

(BTEUP पैटर्न पर अभ्यास पेपर, CSE तीसरा सेमेस्टर: 60 अंक, 3 घंटे। हर पेपर में 5 प्रश्न हैं, प्रति इकाई एक; हर प्रश्न के तीन भाग (a), (b), (c) हैं, हर भाग 6 अंक — हर प्रश्न के कोई दो भाग कीजिए। उत्तर संकेत हर पेपर के बाद दिए गए हैं।)

मॉडल पेपर 1

समय: 3 घंटे · अधिकतम अंक: 60 हर प्रश्न के कोई दो भाग कीजिए। प्रश्न के सभी भाग 6 अंक के हैं।

प्र1. (इकाई 1 — परिचय)

- (a) ऑपरेटिंग सिस्टम को परिभाषित कीजिए। उसके दो लक्ष्यों को एक-एक उदाहरण सहित समझाइए।
- (b) बैच, टाइम-शेयरिंग और रियल-टाइम ऑपरेटिंग सिस्टम का हर एक के एक उपयोग सहित वर्णन कीजिए।
- (c) सिस्टम कॉल क्या है? उसके पाँच प्रकार हर एक के एक उदाहरण सहित गिनाइए।

प्र2. (इकाई 2 — प्रक्रिया प्रबंधन)

- (a) प्रक्रिया अवस्था चित्र बनाइए और हर बदलाव समझाइए।
- (b) प्रक्रियाएँ इस प्रकार आती हैं P1 (AT 0, BT 7), P2 (AT 2, BT 4), P3 (AT 4, BT 1), P4 (AT 5, BT 4)। दोनों Gantt चार्ट दिखाते हुए FCFS और SRTF के अंतर्गत औसत प्रतीक्षा समय गिनिए।
- (c) डेडलॉक की चार आवश्यक शर्तें बताइए। किन्हीं दो के लिए, एक रोकथाम विधि और उसकी लागत दीजिए।

प्र3. (इकाई 3 — मेमोरी प्रबंधन)

- (a) क्रम में मुक्त छिद्र 120, 480, 260, 340, 200 KB हैं; माँगें 300, 140, 420 उसी क्रम में हैं। First Fit और Best Fit के अंतर्गत स्थान (या विफलता) दिखाइए।

(b) पेज आकार 256 और पेज टेबल प्रविष्टि पेज 3 → फ्रेम 6 के साथ, तार्किक एड्रेस 1000 का चरण दर चरण अनुवाद कीजिए। यह भी बताइए कि 5000-बाइट प्रक्रिया को कितने पेज चाहिए और उसका आंतरिक विखंडन कितना है।

(c) संदर्भ स्ट्रिंग 2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 3 फ्रेम के साथ: LRU और Optimal के अंतर्गत पेज फ्रॉन्ट गिनिए।

प्र4. (इकाई 4 — I/O सिस्टम)

(a) एक डिस्क माँग कतार 86, 147, 91, 177, 48, 150, 102, 175 है; हेड 100 पर है और सिलेंडर 0-199 हैं। FCFS और SCAN (हेड बढ़े नंबरों की ओर बढ़ता हुआ, अंत तक जाते हुए) के अंतर्गत कुल हेड गति गिनिए।

(b) RAID स्तर 0, 1 और 5 की तकनीक, न्यूनतम डिस्क और विफलता सहनशीलता सहित व्याख्या कीजिए। चार 1 TB डिस्क के लिए, तीनों में से हर एक के अंतर्गत उपयोगी स्थान दीजिए।

(c) स्वैप स्थान क्या है? स्वैप विभाजन और स्वैप फाइल की तुलना कीजिए, और समझाइए कि OS स्वैप को फाइल सिस्टम के बाहर क्यों प्रबंधित करता है।

प्र5. (इकाई 5 — फाइल प्रबंधन)

(a) फाइल के गुण गिनाइए। अनुक्रमिक और प्रत्यक्ष एक्सेस विधियों को एक-एक उपयुक्त उपयोग सहित समझाइए।

(b) सतत, शृंखलाबद्ध और अनुक्रमित फाइल आवंटन की किन्हीं चार बिंदुओं पर तुलना कीजिए, और उस मरम्मत का नाम बताइए जो FAT शृंखलाबद्ध आवंटन पर लगाता है।

(c) UNIX अनुमति स्ट्रिंग `-rwxr-x---` को पूरी तरह डिकोड कीजिए। फिर इसके लिए स्ट्रिंग बताइए: स्वामी पढ़-लिख सके, समूह पढ़ सके, अन्य कुछ नहीं।

उत्तर संकेत — मॉडल पेपर 1

प्र1. (a) अध्याय 1 §1.1 परिभाषा; सुविधा (हार्डवेयर छिपाना — डबल-क्लिक से फाइल खुलती है) और कुशलता (दूसरे काम के I/O इंटरफ़ारे के दौरान CPU काम करता है)।

(b) अध्याय 1 §1.3 पंक्तियाँ: बैच/वेतन; टाइम-शेयरिंग/कॉलेज सर्वर; रियल-टाइम हार्ड-उड़ान नियंत्रण, सॉफ़्ट-स्ट्रीमिंग।

(c) अध्याय 2 §2.4: user-mode → kernel-mode द्वार; प्रक्रिया नियंत्रण (fork), फाइल (read), उपकरण (ioctl), सूचना (getpid), संचार (pipe)।

प्र2. (a) अध्याय 3 §3.2, चित्र 3.1, I/O पूर्ति पर waiting→ready सहित सभी पाँच तीर।

(b) FCFS क्रम P1, P2, P3, P4: CT 7, 11, 12, 16; WT: 0, 5, 7, 7 → **औसत 4.75**। SRTF: P1 0-2 चलती है; P2 (4) बनाम P1 शेष 5 → P2 2-4; P3 (1) 4 पर आती है → 4-5 चलती है; फिर P2 शेष 2 → 5-7; P4 (4) बनाम P1 (5) → P4 7-11; P1 11-16। CT: P3 5, P2 7, P4 11, P1 16; WT = CT - AT - BT: P1 9, P2 1, P3 0, P4 2 → **औसत 3**।

(c) अध्याय 6 §6.2 सूची; §6.3 से रोकथाम जोड़े (स्पूलिंग प्रभाव में परस्पर अपवर्जन तोड़ता है; सब-कुछ-एक-साथ माँगना धारण-और-प्रतीक्षा तोड़ता है — खाली धारण किए संसाधन; संसाधन क्रम वृत्तीय प्रतीक्षा तोड़ता है — सबके लिए एक वैश्विक क्रम)।

प्र3. (a) First Fit: 300 → 480 (शेष 180); 140 → 180 (शेष 40); 420 → कोई छिद्र नहीं बैठता (छिद्र 120, 40, 260, 340, 200) — विफलता। Best Fit: 300 → 340 (शेष 40); 140 → 200 (शेष 60); 420 → 480 (शेष 60) — सब रखे गए।

(b) $p = 1000 \div 256 = 3$, $d = 1000 - 768 = 232$; भौतिक = $6 \times 256 + 232 = 1768$ । पेज चाहिए = $\lceil 5000/256 \rceil = 20$ (19.53...); आंतरिक विखंडन = $20 \times 256 - 5000 = 120$ बाइट।

(c) LRU: केवल दोहराया गया 2 (तीसरा संदर्भ) hit होता है; अन्य नौ संदर्भ फ़ॉल्ट करते हैं → **9 फ़ॉल्ट**। (खोजिए: 2, 3, 2, 1 के बाद फ्रेम {2, 1} रखते हैं; 5, 2 को निकालता है, फिर 2, 1 को, 4, 5 को, 5, 2 को, 3, 4 को, 2, 5 को — स्ट्रिंग ठीक वही माँगती रहती है जो LRU ने अभी निकाला।) Optimal: फ़ॉल्ट 2, 3, 1, 5, 4, 3, 2 पर → **7 फ़ॉल्ट** (स्थिति 3 और 6 पर 2 के संदर्भ, और स्थिति 8 पर 5 का संदर्भ, hit होते हैं)। एक छोटी स्ट्रिंग पर 9 और 7 के बीच का अंतर भविष्य न जानने की LRU की कीमत है।

प्र4. (a) FCFS: 100→86 (14) →147 (61) →91 (56) →177 (86) →48 (129) →150 (102) →102 (48) →175 (73) = **569**। SCAN (ऊपर, अंत 199 तक): 102, 147, 150, 175, 177 सेवित, फिर 199, फिर नीचे: 91, 86, 48। गति: $2 + 45 + 3 + 25 + 2 + 22 + 108 + 5 + 38 = 250$ ।

(b) अध्याय 11 §11.3 तालिका पंक्तियाँ; 4×1 TB: RAID 0 → 4 TB उपयोगी, 0 सहता है; RAID 1 (चार डिस्क का दर्पण समूह दो दर्पण युग्मों के रूप में — कठोर नामकरण में यह RAID 10 है) → 2 TB उपयोगी, प्रति युग्म एक सहता है; RAID 5 → 3 TB उपयोगी, 1 सहता है।

(c) अध्याय 11 §11.2: परिभाषा; विभाजन = कच्चा/निश्चित/तेज़, फाइल = लचीली/धीमी; गति के लिए और इसलिए फाइल सिस्टम को बायपास करना क्योंकि सामग्री अस्थायी है (कोई नाम, साझा करना या बैकअप नहीं)।

प्र5. (a) अध्याय 12 §12.1 गुण सूची; §12.2: अनुक्रमिक — CCTV/लॉग; प्रत्यक्ष — ब्लॉक-नंबर विचार के साथ डेटाबेस रिकॉर्ड लाना।

(b) अध्याय 13 §13.5 तालिका (कोई चार पंक्तियाँ); FAT सूचकों को एक कैश-योग्य तालिका में एकत्र

करती है।

(c) स्वामी rwx (पढ़ना, लिखना, निष्पादित), समूह r-x (पढ़ना, निष्पादित), अन्य --- (कुछ नहीं); अग्रणी '-' = साधारण फाइल। आवश्यक स्ट्रिंग: **-rw-r-----**।

मॉडल पेपर 2

समय: 3 घंटे • अधिकतम अंक: 60 हर प्रश्न के कोई दो भाग कीजिए। प्रश्न के सभी भाग 6 अंक के हैं।

प्र1. (इकाई 1 — परिचय)

- (a) कर्नेल क्या है? मोनोलिथिक और माइक्रोकर्नेल डिज़ाइन की चार बिंदुओं पर तुलना कीजिए।
- (b) UNIX/Linux की परतदार संरचना समझाइए, हार्डवेयर से उपयोगकर्ता तक सभी परतों के नाम देते हुए।
- (c) "टाइम-शेयरिंग सिस्टम मल्टीप्रोग्रामिंग का तार्किक विस्तार है।" लगभग आठ लाइनों में पुष्टि कीजिए।

प्र2. (इकाई 2 — प्रक्रिया प्रबंधन)

- (a) सेमाफोर क्या है? wait और signal को सटीक परिभाषित कीजिए, और तीन सेमाफोरों का उपयोग करके उत्पादक-उपभोक्ता समस्या हल कीजिए, हर एक का प्रारंभिक मान और काम बताते हुए।
- (b) एक सिस्टम में 4 प्रक्रियाएँ और 3 संसाधन प्रकार हैं जिनके योग $A = 9$, $B = 6$, $C = 6$ हैं। आवंटन: $P_0 (1, 1, 0)$, $P_1 (2, 0, 1)$, $P_2 (1, 0, 2)$, $P_3 (0, 2, 0)$ । Max: $P_0 (5, 3, 2)$, $P_1 (3, 2, 2)$, $P_2 (4, 1, 3)$, $P_3 (2, 4, 2)$ । Available और Need खोजिए, और तय कीजिए कि अवस्था सुरक्षित है या नहीं, यदि है तो सुरक्षित क्रम देते हुए।
- (c) साझा-काउंटर उदाहरण के साथ रेस कंडीशन समझाइए, और क्रिटिकल-सेक्शन हल की तीन शर्तें बताइए।

प्र3. (इकाई 3 — मेमोरी प्रबंधन)

- (a) आंतरिक और बाहरी विखंडन को एक-एक उदाहरण सहित अलग कीजिए, और हर एक से पीड़ित एक आवंटन योजना का नाम बताइए।
- (b) सेगमेंटेशन समझाइए: एड्रेस भाग, सेगमेंट टेबल की सामग्री, और लिमिट जाँच। सेगमेंट 1: बेस 4200, लिमिट 600 दिए होने पर (सेगमेंट 1, ऑफ़सेट 350) का अनुवाद कीजिए — और बताइए ऑफ़सेट 700 के लिए क्या होता है।
- (c) वर्चुअल मेमोरी क्या है? थ्रैशिंग, कार्यशील समूहों के संदर्भ में उसका कारण, और दो उपचार समझाइए।

प्र4. (इकाई 4 — I/O सिस्टम)

- (a) हार्ड डिस्क की भौतिक संरचना (प्लैटर, ट्रैक, सेक्टर, सिलेंडर) का वर्णन कीजिए और 16 सतहों, प्रति सतह 2048 ट्रैक, प्रति ट्रैक 128 सेक्टर और प्रति सेक्टर 512 बाइट वाली डिस्क की क्षमता GiB में गिनिए।
- (b) होस्ट-अटैच्ड स्टोरेज, NAS और SAN को इस आधार पर अलग कीजिए कि हर एक क्या निर्यात करता है और किस मार्ग पर।
- (c) एक माँग कतार 30, 90, 60, 120, 10 है, हेड 50 पर है (सिलेंडर 0-199)। SSTF और C-SCAN (ऊपर बढ़ता हुआ; वापसी कूद गिनते हुए) के अंतर्गत कुल हेड गति गिनिए।

प्र5. (इकाई 5 — फाइल प्रबंधन)

- (a) डायरेक्टरी संरचनाओं के विकास को एकल-स्तरीय से वृक्ष-संरचित तक खोजिए, हर चरण ने जो समस्या हल की वह बताते हुए। अपने उत्तर में निरपेक्ष बनाम सापेक्ष पथ शामिल कीजिए।
- (b) चार मुक्त-स्थान प्रबंधन योजनाओं को संक्षेप में समझाइए। मुक्त ब्लॉक {5, 6, 7, 20, 21, 40} के लिए, गिनती योजना के अंतर्गत रिकॉर्ड लिखिए।
- (c) फाइल सिस्टम माउंटिंग क्या है? एक पेन ड्राइव उदाहरण से समझाइए, माउंट बिंदु और सुरक्षित निष्कासन सहित।

उत्तर संकेत — मॉडल पेपर 2

- प्र1. (a) अध्याय 2 §2.3 परिभाषा + चार-पंक्ति तुलना तालिका (अंदर सेवाएँ, गति, विश्वसनीयता, उदाहरण Linux/MINIX)।
- (b) हार्डवेयर → कर्नेल → सिस्टम-कॉल इंटरफ़ेस → लाइब्रेरी/शेल/उपयोगिताएँ → उपयोगकर्ता और एप्लिकेशन (अध्याय 2 §2.5)।
- (c) QB इकाई 1, खंड C प्र3 के मुख्य बिंदु: वही बहु-काम आधार, टाइमर-आधारित बदलाव जोड़ा गया, नई विशेषता के रूप में बातचीत।

- प्र2. (a) अध्याय 5 §5.4 परिभाषा (atomicता बताई गई) और §5.5: mutex = 1 (बफ़र की रखवाली), empty = N (मुक्त स्थान), full = 0 (वस्तुएँ); उत्पादक: wait(empty), wait(mutex), जोड़ना, signal(mutex), signal(full); उपभोक्ता full/empty बदले हुए दर्पण रूप में।
- (b) आवंटित योग: A = 4, B = 3, C = 3 → **Available = (5, 3, 3)**। Need = Max - Allocation: P0 (4, 2, 2), P1 (1, 2, 1), P2 (3, 1, 1), P3 (2, 2, 2)। Work (5, 3, 3) से सुरक्षा: हर Need पहले ही पास में बैठता है — जैसे, P1 (1,2,1) → Work (7, 3, 4); P0 (4,2,2) → (8, 4,

4); P2 (3,1,1) → (9, 4, 6); P3 (2,2,2) → (9, 6, 6)। **सुरक्षित**, एक क्रम P1, P0, P2, P3 (अन्य मौजूद हैं)।

(c) अध्याय 5 §5.2 काउंटर कहानी (4 → 6 के बजाय 5) और §5.3 की तीन शर्तें, एक-एक लाइन।

प्र3. (a) अध्याय 7 के उदाहरणों सहित अध्याय 9 §9.3: आंतरिक — 250 KB निश्चित विभाजन में 60 KB प्रक्रिया / अंतिम पेज पूँछ (निश्चित विभाजन, पेजिंग); बाहरी — 959 KB मुक्त के बीच विफल होती 426 KB माँग (परिवर्तनशील विभाजन, सेगमेंटेशन)।

(b) अध्याय 8 §8.3: एड्रेस = सेगमेंट संख्या + ऑफ़सेट; टेबल बेस + लिमिट रखती है; जाँच ऑफ़सेट < लिमिट: 350 < 600 ✓ → भौतिक = 4200 + 350 = **4550**; ऑफ़सेट 700 ≥ 600 → **फ़ॉल्ट** (सुरक्षा काम कर रही है)।

(c) अध्याय 9 §9.4: परिभाषा (वर्चुअल स्पेस पर डिमांड पेजिंग); संयुक्त कार्यशील समूह फ्रेम से अधिक होने पर थ्रैशिंग → फ़ॉल्ट तूफ़ान और CPU पतन; उपचार — प्रति कार्यशील समूह फ्रेम की गारंटी, या मल्टीप्रोग्रामिंग की मात्रा घटाना।

प्र4. (a) अध्याय 10 §10.2 रचना; क्षमता = $16 \times 2048 \times 128 \times 512$ बाइट = 2,147,483,648 बाइट = **2 GiB**।

(b) अध्याय 10 §10.3: होस्ट-अटैच्ड कुछ निर्यात नहीं करता (एक होस्ट से तारबद्ध बस); NAS साधारण LAN पर फाइलें निर्यात करता है; SAN समर्पित स्टोरेज नेटवर्क पर कच्चे ब्लॉक निर्यात करता है।

(c) 50 से SSTF: 60 (10), 90 (30), 120 (30), 30 (90), 10 (20) = **180**। C-SCAN ऊपर: 60, 90, 120 सेवित, 199 तक, 0 पर कूद, 10, 30 सेवित: गति 10 + 30 + 30 + 79 (199 तक) + 199 (कूद) + 10 + 20 = **378**।

प्र5. (a) QB इकाई 5, खंड C प्र2 के मुख्य बिंदु (एकल → द्वि-स्तरीय → वृक्ष), निरपेक्ष उदाहरण के रूप में /usr/bin/python3 और बताए गए कार्यशील डायरेक्टरी से एक सापेक्ष रूप सहित।

(b) अध्याय 13 §13.6 एक-एक लाइन (बिटमैप, श्रृंखलाबद्ध सूची, समूहीकरण, गिनती); गिनती रिकॉर्ड: **(5, 3), (20, 2), (40, 1)**।

(c) अध्याय 12 §12.4: माउंटिंग किसी फाइल सिस्टम की जड़ को जीवित वृक्ष में एक माउंट बिंदु (मौजूद डायरेक्टरी) पर जोड़ता है; पेन ड्राइव /media/usb पर या ड्राइव अक्षर पर; अनमाउंट प्रतीक्षारत लेखन प्लश करता है — जल्दी खींचना भ्रष्ट कर सकता है, इसलिए सुरक्षित निष्कासन।



पिछले वर्षों के पेपर — बोर्ड ने असल में क्या पूछा

(पहले यह पढ़िए। यह किताब **BTEUP पॉलिटेक्निक डिप्लोमा — Computer Science व IT** के छात्रों के लिए है। 2025 तक के वर्षों में यह विषय बोर्ड की पुरानी योजना में उसी डिप्लोमा के **Operating System, कोड 2207, चौथे सेमेस्टर** के रूप में चला: **2½ घंटे में 50 अंक** — पाँच प्रश्न, हर एक के कोई दो भाग, प्रति भाग 5 अंक — Linux पर एक सातवीं इकाई के साथ जो आज के पाठ्यक्रम ने हटा दी है। वर्तमान योजना से यह **विषय 3.1, तीसरा सेमेस्टर, 60 अंक, 3 घंटे** है। नीचे इसी डिप्लोमा के असली पेपर हैं, ठीक जैसे पूछे गए। हर प्रश्न के सबसे आगे उसका टैग है — साल, सेमेस्टर और प्रश्न संख्या — और हर पेपर के बाद हमारे उत्तर संकेत हैं।)

इस डिप्लोमा के पेपरों में दोहराए गए प्रश्न

इस डिप्लोमा विषय (कोड 2207) के दो असली पेपर उपलब्ध हैं — मई 2025 का, और एक और 2207 पेपर जिसकी उपलब्ध प्रति पर उसका साल नहीं है। ये प्रश्न **दोनों** में आते हैं — इस विषय के बोर्ड के सिद्ध पसंदीदा:

- **[मई 2025 · प्र1(b)] + [साल छपा नहीं · प्र1(a)]** — ऑपरेटिंग सिस्टम के विभिन्न कार्य समझाइए। उत्तर: अध्याय 2, §2.2।
- **[मई 2025 · प्र3(b)] + [साल छपा नहीं · प्र3(b)]** — पेजिंग समझाइए, उदाहरण सहित / टिप्पणी के रूप में। उत्तर: अध्याय 8, §8.2।
- **[मई 2025 · प्र4(b)] + [साल छपा नहीं · प्र3(b)]** — Segmentation पर टिप्पणी / व्याख्या। उत्तर: अध्याय 8, §8.3।
- **[मई 2025 · प्र4(a)] + [साल छपा नहीं · प्र4(b)]** — Spooling: टिप्पणी / buffering और spooling विस्तार से। उत्तर: नीचे पेपर 1, प्र4(a) संकेत।
- **[मई 2025 · प्र3(c)] + [साल छपा नहीं · प्र2(c)]** — डेडलॉक: रोकथाम / आवश्यक शर्तें और निपटारा। उत्तर: अध्याय 6, §§6.2–6.3।
- **[मई 2025 · प्र5(a)] + [साल छपा नहीं · प्र5(b)]** — कोई पाँच Linux कमांड उदाहरण / कार्यों सहित। पुराने पाठ्यक्रम की Linux इकाई।

- [मई 2025 • प्र5(c)] + [साल छपा नहीं • प्र5(c)] — Shell script: क्या होती है / उदाहरण सहित समझाइए। पुराने पाठ्यक्रम की Linux इकाई।
- एक नामित शेड्यूलिंग एल्गोरिदम, हर बार परीक्षित — उदाहरण सहित Round Robin [मई 2025 • प्र2(b)]; शेड्यूलिंग एल्गोरिदम के प्रकार और एक FCFS आंकिक प्रश्न [साल छपा नहीं • प्र1(b), प्र2(b)]।

पेपर 1 — Operating System (कोड 2207), मई 2025, चौथा सेमेस्टर

✓ बोर्ड के पेपर से सत्यापित

समय: 2:30 घंटे • अधिकतम अंक: 50 • सभी प्रश्न कीजिए।

प्र1 — कोई दो भाग कीजिए [2×5=10]:

- [2025 • सेम 4 • प्र1(a)] ऑपरेटिंग सिस्टम को परिभाषित कीजिए और उसके प्रकार समझाइए।
- [2025 • सेम 4 • प्र1(b)] ऑपरेटिंग सिस्टम के विभिन्न कार्य समझाइए।
- [2025 • सेम 4 • प्र1(c)] सिस्टम सॉफ्टवेयर और एप्लिकेशन सॉफ्टवेयर के बीच अंतर बताइए।

प्र2 — कोई दो भाग कीजिए [2×5=10]:

- [2025 • सेम 4 • प्र2(a)] प्रोसेस स्टेट क्या है? विभिन्न प्रोसेस स्टेट समझाइए।
- [2025 • सेम 4 • प्र2(b)] उपयुक्त उदाहरण सहित Round Robin (RR) शेड्यूलिंग एल्गोरिदम समझाइए।
- [2025 • सेम 4 • प्र2(c)] अंतर-प्रक्रिया संचार की विस्तार से चर्चा कीजिए।

प्र3 — कोई दो भाग कीजिए [2×5=10]:

- [2025 • सेम 4 • प्र3(a)] आंतरिक और बाहरी विखंडन के बीच अंतर बताइए।
- [2025 • सेम 4 • प्र3(b)] उदाहरण सहित पेजिंग समझाइए।
- [2025 • सेम 4 • प्र3(c)] डेडलॉक रोकथाम की विधियाँ समझाइए।

प्र4 — निम्नलिखित में से किन्हीं दो पर टिप्पणी लिखिए [2×5=10]:

- [2025 • सेम 4 • प्र4(a)] Spooling
- [2025 • सेम 4 • प्र4(b)] Segmentation
- [2025 • सेम 4 • प्र4(c)] फाइल सिस्टम के प्रकार

प्र5 — कोई दो भाग कीजिए [2×5=10] (पुराने पाठ्यक्रम की Linux इकाई; आज के विषय 3.1 में नहीं है — यहाँ इसलिए रखा है क्योंकि असली पेपर में यह है और पुरानी योजना के बैक-पेपर छात्रों को इसकी अब भी आवश्यकता है):

- **[2025 · सेम 4 · प्र5(a)]** उदाहरण सहित किन्हीं पाँच Linux कमांड की चर्चा कीजिए।
- **[2025 · सेम 4 · प्र5(b)]** Linux सिस्टम आवश्यकताएँ क्या हैं?
- **[2025 · सेम 4 · प्र5(c)]** Shell script से आप क्या समझते हैं?

उत्तर संकेत — पेपर 1 (मई 2025)

- **प्र1(a)** [2025 · सेम 4 · प्र1(a)] अध्याय 1, §1.1 (संपूर्ण परिभाषा) और §1.3 (प्रकार तालिका)।
प्र1(b) अध्याय 2, §2.2: पाँच कार्य। **प्र1(c)** अध्याय 1 का सिस्टम-सॉफ्टवेयर परिवार बनाम चित्र 1.1 के एप्लिकेशन प्रोग्राम।
- **प्र2(a)** अध्याय 3, §3.2 चित्र 3.1 सहित — चित्र बनाइए; बोर्ड उसे पुरस्कृत करता है। **प्र2(b)** अध्याय 4, §4.4 (डेटासेट A की RR चाल तैयार है)। **प्र2(c)** अध्याय 5, §5.1।
- **प्र3(a)** अध्याय 9, §9.3। **प्र3(b)** अध्याय 8, §8.2 (753 → 453)। **प्र3(c)** अध्याय 6, §6.3।
- **प्र4(a) Spooling:** प्रिंट काम पहले एक डिस्क कतार (spool) में लिखे जाते हैं और बारी से छपते हैं, इसलिए प्रोग्राम धीमे प्रिंटर का इंतज़ार कभी नहीं करते और प्रिंटर धारण किए बिना साझा होता है — अध्याय 6, §6.3; रोज़ की तस्वीर अध्याय 1 की प्रिंट कतार है। **प्र4(b)** अध्याय 8, §8.3। **प्र4(c)** अध्याय 13, §13.7।
- **प्र5 (Linux, पुराना पाठ्यक्रम):** (a) `ls`, `cd`, `mkdir`, `cp`, `cat`, `rm`, `pwd`, `chmod` में से कोई पाँच, एक-पंक्ति काम और उदाहरण सहित, जैसे लैब प्रयोग 1 में अभ्यास किया गया।
(b) एक विशिष्ट Linux वितरण लगभग किसी भी आधुनिक PC पर स्थापित हो जाता है: 64-बिट प्रोसेसर, लगभग 2 GB या अधिक RAM, सामान्य desktop वितरणों के लिए लगभग 20–25 GB मुक्त डिस्क (न्यूनतम स्थापना को कहीं कम चाहिए) — बोर्ड का बिंदु: Linux मामूली हार्डवेयर पर अच्छी तरह चलता है।
(c) Shell script shell कमांड की क्रम में एक सादा पाठ फाइल है, जिसे shell एक-एक करके चलाता है (`chmod +x file.sh`, फिर `./file.sh`)।

पेपर 2 — Operating System (कोड 2207), उपलब्ध प्रति पर साल छपा नहीं है ✓ असली बोर्ड पेपर

(इस पेपर पर इसी डिप्लोमा विषय का छपा **कोड नं. 2207** है और वही 50-अंक / 2½-घंटे पैटर्न है। सार्वजनिक रूप से उपलब्ध प्रति एक-पृष्ठ preview है जिस पर साल और सेमेस्टर दिखाई नहीं देते, इसलिए इसके लिए कोई साल दावा नहीं किया जाता। इसके प्रश्न ठीक जैसे पूछे गए वैसे छपे हैं, और इसका आंकिक प्रश्न नीचे हल किया गया है।)

समय: 2:30 घंटे • अधिकतम अंक: 50 • न्यूनतम अंक: 17 • सभी प्रश्न कीजिए, हर एक के कोई दो भाग।

प्र1:

- [साल छपा नहीं • प्र1(a)] ऑपरेटिंग सिस्टम के कार्य समझाइए।
- [साल छपा नहीं • प्र1(b)] प्रोसेस शेड्यूलिंग एल्गोरिदम के विभिन्न प्रकार समझाइए।
- [साल छपा नहीं • प्र1(c)] संक्षिप्त टिप्पणी लिखिए: i) Context switch ii) Process Control Block (PCB).

प्र2:

- [साल छपा नहीं • प्र2(a)] उदाहरण सहित Pre-emptive और Non pre-emptive एल्गोरिदम के बीच अंतर बताइए।
- [साल छपा नहीं • प्र2(b)] आंकिक: FCFS एल्गोरिदम का उपयोग करके निम्नलिखित प्रक्रियाओं के लिए औसत प्रतीक्षा समय और औसत टर्नअराउंड समय खोजिए — P0: आगमन 1, बर्स्ट 4; P1: आगमन 0, बर्स्ट 12; P2: आगमन 2, बर्स्ट 3 (बर्स्ट समय ms में)।
- [साल छपा नहीं • प्र2(c)] डेडलॉक के लिए आवश्यक शर्तें और डेडलॉक से निपटने की विधियाँ समझाइए।

प्र3:

- [साल छपा नहीं • प्र3(a)] ऑपरेटिंग सिस्टम में मेमोरी प्रबंधन को विस्तार से समझाइए। साथ ही, तार्किक एड्रेस स्पेस और भौतिक एड्रेस स्पेस की चर्चा कीजिए।
- [साल छपा नहीं • प्र3(b)] संक्षिप्त टिप्पणी लिखिए: i) Paging ii) Segmentation.
- [साल छपा नहीं • प्र3(c)] वर्चुअल मेमोरी को विस्तार से समझाइए।

प्र4:

- [साल छपा नहीं • प्र4(a)] सिस्टम कॉल और सिस्टम कॉल के प्रकारों की चर्चा कीजिए।

- [साल छपा नहीं • प्र4(b)] Buffering और spooling की विस्तार से चर्चा कीजिए।
- [साल छपा नहीं • प्र4(c)] I/O प्रबंधन कार्य समझाइए।

प्र5:

- [साल छपा नहीं • प्र5(a)] Linux ऑपरेटिंग सिस्टम का अवलोकन लिखिए। इसकी संरचना और विशेषताएँ समझाइए।
- [साल छपा नहीं • प्र5(b)] निम्नलिखित Linux कमांड के कार्य लिखिए: i) rmdir ii) pwd iii) grep iv) chmod v) kill.
- [साल छपा नहीं • प्र5(c)] उपयुक्त उदाहरणों सहित shell script और shell कमांड समझाइए।

पेपर 2 का आंकिक प्रश्न, हल किया गया

FCFS आगमन क्रम में सेवा करता है: P1 पहले आती है (समय 0), फिर P0 (समय 1), फिर P2 (समय 2)। Gantt: **P1 0-12 चलती है, P0 12-16, P2 16-19।** पूर्ति समय: P1 = 12, P0 = 16, P2 = 19। टर्नअराउंड = पूर्ति - आगमन: P1 = 12, P0 = 15, P2 = 17 → **औसत टर्नअराउंड समय = $44/3 \approx 14.67$ ms।** प्रतीक्षा = टर्नअराउंड - बर्स्ट: P1 = 0, P0 = 11, P2 = 14 → **औसत प्रतीक्षा समय = $25/3 \approx 8.33$ ms।** सबक, FCFS के साथ हमेशा की तरह: P2 का छोटा 3 ms काम P1 के लंबे काम के पीछे 14 ms प्रतीक्षा करता है — convoy प्रभाव (अध्याय 4, §4.2)। पेपर 2 के शेष संकेत: प्र1 — अध्याय 2 §2.2, अध्याय 4 अवलोकन, अध्याय 3 §§3.4-3.5; प्र2(a) अध्याय 4 §4.1, प्र2(c) अध्याय 6 §§6.2-6.3; प्र3 — अध्याय 7 §7.1, अध्याय 8 §§8.2-8.3, अध्याय 9 §9.4; प्र4 — अध्याय 2 §2.4 (सिस्टम कॉल), buffering/spooling पेपर 1 प्र4(a) की तरह, अध्याय 2 §2.2 का I/O कार्य; प्र5 — पुरानी Linux इकाई: **rmdir** खाली डायरेक्टरी हटाता है, **pwd** वर्तमान डायरेक्टरी छापता है, **grep** फाइलों में पाठ खोजता है, **chmod** अनुमतियाँ बदलता है, **kill** प्रक्रिया को उसके PID से समाप्त करता है; shell script पेपर 1 प्र5(c) की तरह।

2024 की परीक्षाएँ — हुई, पेपर सार्वजनिक नहीं

बोर्ड की आधिकारिक परीक्षा योजनाएँ पुष्टि करती हैं कि इस डिप्लोमा का पेपर 2024 में दो बार हुआ: **1 जुलाई 2024** (सम-सेमेस्टर मुख्य परीक्षा, चौथा सेमेस्टर) और **16 जनवरी 2024** (बैक व विशेष बैक परीक्षा), दोनों कोड 2207 के रूप में 2½ घंटे के स्लॉट में। प्रश्न पत्र स्वयं किसी सार्वजनिक भंडार में उपलब्ध नहीं हैं जहाँ तक हम पहुँच सके, इसलिए — उनकी सामग्री का अनुमान लगाने के बजाय — यह

बैंक 2024 के लिए कुछ नहीं छापता। यदि आपके संस्थान का पुस्तकालय बोर्ड पेपर रखता है, तो माँगने के लिए वही दो तारीखें हैं। 2020, 2021 और 2022 के पेपर भी किसी विश्वसनीय स्रोत से सत्यापित नहीं हो सके, और कोई छपा नहीं है।

ये पेपर आपको क्या सिखाते हैं (ईमानदार पैटर्न)

इस डिप्लोमा के असली पेपरों में, वही ढाँचा दिखता है: OS मूल बातों, प्रकारों और कार्यों पर एक प्रश्न; प्रक्रियाओं और शेड्यूलिंग पर एक — हर बार एक नामित एल्गोरिदम के साथ; मेमोरी (पेजिंग, विखंडन) पर एक जिसके बगल में डेडलॉक; क्लासिक छोटे विषयों पर संक्षिप्त टिप्पणियाँ (buffering, spooling, segmentation); फाइल-सिस्टम और सिस्टम-कॉल प्रश्न; और बंद करने के लिए पुरानी Linux इकाई। पहले वे स्तंभ तैयार कीजिए — वे इस बैंक की इकाइयाँ 1-3 और हल किए गए आंकिक प्रश्न हैं — और इस विषय का कोई पेपर, पुरानी योजना का हो या नई का, आपको चौंका नहीं सकता।

